

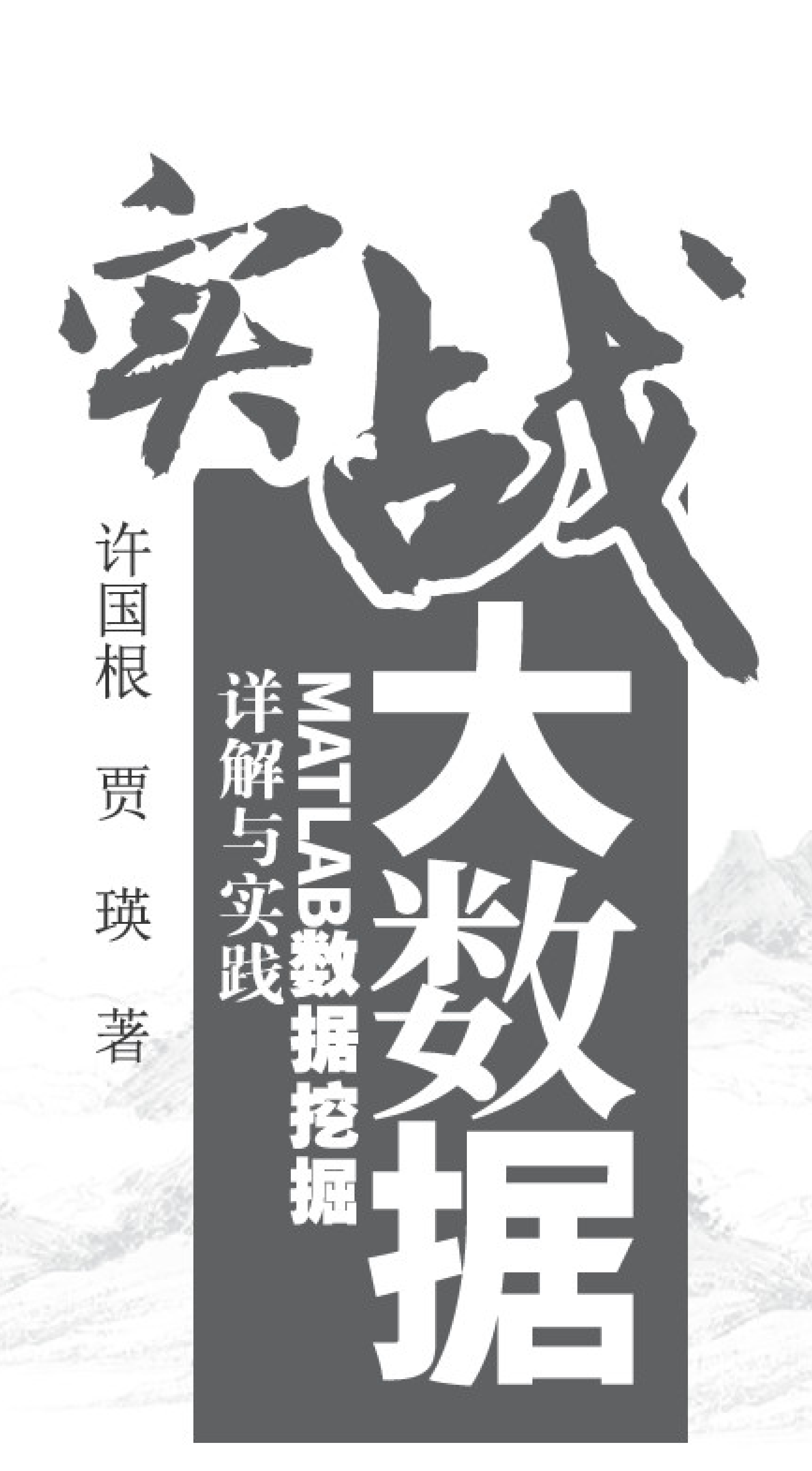
实战大数据

MATLAB数据挖掘详解与实践

深入探究数据挖掘之有效工具
深入讲解数据挖掘各个环节技术
利用数据挖掘算法和MATLAB进行数据分析

许国根 贾瑛 著

清華大學出版社



实战大数据

MATLAB数据挖掘详解与实践

许国根 贾 瑛 著



清華大學出版社
北 京

内 容 简 介

大数据时代，我们需要对各种海量数据进行筛选、清洗、挖掘，在这个过程中，获取有效数据的方式方法和模型算法成为了整个数据挖掘过程的重点，MATLAB 作为一个数据挖掘工具，如何正确和准确地使用它成为了重中之重。

针对实际应用数据挖掘技术的要求，本书既介绍了数据挖掘的基础理论和技术，又较为详细地介绍了各种算法以及 MATLAB 程序。本书共分 4 篇，分别介绍了数据挖掘的基本概念、技术与算法以及应用实例。期望通过大量的实例分析帮助广大读者掌握数据挖掘技术，并应用于实际的研究中，提高对海量数据信息的处理及挖掘能力。本书针对性和实用性强，具有较高的理论和实用价值。

本书作者就职于部队高校，专攻数据挖掘，并应用于大量实际项目，本书同时得到了国内著名数据挖掘公司的技术支持，很多案例来自实际项目。

本书可作为高等院校计算机工程、信息工程、生物医学工程、化学、环境、经济、管理等学科的研究生、本科生的教材或教学参考书，亦可作为企事业单位管理者、信息分析人员、市场营销人员和研究与开发人员的参考资料。

图书在版编目(CIP)数据

实战大数据——MATLAB 数据挖掘详解与实践 / 许国根编著. --北京：清华大学出版社，2017

ISBN 978-7-302-45101-3

Ⅰ. ①实 … Ⅱ. ①许 … Ⅲ. ①MATLAB 软件 Ⅳ. ① TP317

中国版本图书馆 CIP 数据核字（2016）第 227229 号

责任编辑：栾大成
装帧设计：杨玉芳
责任校对：徐俊伟
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，　http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　　邮　　编：100084
社 总 机：010-62770175　　　邮　　购：010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：188mm×230mm　　**印　张：**35.25　　**字　数：**904 千字
版　　次：2017 年 7 月第 1 版　　**印　次：**2017 年 7 月第 1 次印刷
印　　数：1~3500
定　　价：99.00 元

产品编号：058297-01

前　言

计算机技术和通信技术的迅猛发展将人类社会带入了信息时代，在最近十几年里数据库中存储的数据量急剧增大。网络经济、注意力经济等新概念的提出，以其巨大的社会效益和极富挑战与机遇的内涵，成为信息科学引人注目的研究课题。大量的信息在给人们提供方便的同时也带来了一系列的问题，由于信息量过大，超出了人们掌握、理解信息的能力，因而给正确运用信息带来了困难。诸如信息过量、难以消化；信息真假难以辨识；信息安全难以保证；信息形式不一致，难以统一处理，等等，造成了“数据丰富，知识贫乏”。

决策者很难从海量的数据中提取出有价值的知识，促使人们产生了对数据分析工具的强烈需求，如何理解已有的历史数据并用以预测未来的行为，如何从这些海量数据中发现信息，变被动的数据为主支的知识，如何快速、准确地获得有价值的网络信息和网络服务，为用户提供重要的、未知的信息或知识、指导政府决策、企业决策、获取更大的经济效益和社会效益，这些都迫使人们去寻找新的、更为有效的数据分析手段，对各种“数据矿藏”进行有效的挖掘以发挥其应用潜能，20 世纪 80 年代后期至今，数据挖掘正是在这样的应用需求背景下产生并迅速发展起来的，它是开发信息资源的一套科学方法、算法及软件工具和环境。

利用数据分析工具所获取的信息和知识，可以广泛地用于商务管理、生产控制、市场分析、工程设计和科学研究与探索等诸多方面。它不仅是一个重要的研究领域，而且在现实世界具有重大的潜在应用价值。

数据挖掘技术涉及人工智能的机器学习、模式识别、数据库与数据仓库、统计学、可视化图形学等各个领域，吸引了大批学者从事学术研究和工具产品的开发。20 世纪 90 年代中后期，在国外数据挖掘已经形成高潮，我国研究数据挖掘的学者数量也在迅速增长。

由于数据挖掘是一门新兴的、正在不断发展的学科，其理论仍然不成熟，没有完善的理论体系，很多人在没有经历数据挖掘成熟应用项目的情况下，投入到这方面的技术探索与实践工作中来，效果不明显，使得他们对于数据挖掘的许多方面都在打问号，认为数据挖掘是虚的，是炒作。实际上数据挖掘与很多技术一样是一项很实用的技术，它必然会给各行各业的工作带来高效益和高效率。

从应用的角度看，数据挖掘是一个工具，为了很好地应用数据挖掘，首先要了解数据挖掘，尽量掌握数据挖掘的技术和方法，须知道什么时候应该使用何种数据挖掘技术，了解模型内部机制，这样才可以知道如何有效地准备建模所用的数据集，以及如何使用不同参数来改进模型的输出结果。现在有关数据挖掘的书籍越来越多，但这些书籍理论性太强，满篇数学公式，即使给出算法，也只是伪代码，看不到源代码以及算法的实际效果和各种算法的对比结果，而且应用实例很少，这往往使大多数读者感到困惑，让人难以理解，不知如何下手。有时虽然可以借助现在的专业计算机软件来完成数据挖掘工作，但因涉及知识产权保护和价格等因素，不可能每个需要进行数据挖掘的读者都能拥有此类软件。对大多数读者而言，目前确实还缺少一本具有较强系统性、可比性和实用性的有关数据挖掘的参考书。基于这点考虑，作者撰写了本书，向读者介绍各种数

据挖掘技术、方法及基于 MATLAB 的具体算法。想通过系统的介绍和实例分析，让众多的读者不仅具备数据挖掘的理论，而且能掌握数据挖掘应用方法，可以在各自的学科实际研究中予以应用，也使数据挖掘技术更易于使用和开发。

本书按照理论基础、实现步骤、实例三部分内容进行阐述，避免空洞的理论说教，着重介绍应用实例，具有较强的指导性和实用性，使读者不至于面对如此丰富的理论和方法无所适从，而是通过了解各种算法的实现思路和方法，体会算法源代码的意义，这样即使所举的实例不属于读者从事的学科，也能举一反三，掌握数据挖掘技术并应用于自己从事的科学研究中。

本书分为 4 篇，每篇涵盖的内容比较广泛，既有对数据挖掘概念的讨论，也有对数据挖掘技术和原理的介绍，而且编写了大量的实例，并给出了相应的程序。第 1 篇介绍数据挖掘的相关概念以及在多个领域中的应用情况；第 2 篇介绍数据挖掘算法，包括人工神经网络、决策树、遗传算法、关联分析、统计分析方法、支持向量机及一些聚类算法；第 3 篇介绍与数据挖掘相关的一些技术，包括数据仓库、模糊集理论、粗糙集技术、目标优化技术、可视化技术、公式发现、多媒体数据挖掘技术及 Web 数据挖掘技术；第 4 篇介绍数据挖掘具体应用实例，重点介绍数据预处理技术、聚类、分类、预测、关联规则分析、时间序列分析等方法。

本书的出版得到了清华大学出版社的大力支持，编辑栾大成为本书内容等许多方面提出了宝贵的意见。书中参考了许多学者的研究成果，在此一并表示衷心的感谢！

由于数据挖掘的内容非常丰富，所涉及的学科也较多，且限于作者学识水平，书中难免存在缺点、错误以及疏漏，敬请读者批评斧正。

本书为读者提供实例中给出的 MATLAB 程序，需要者可登录清华大学出版社网站，注册之后进行下载。读者反馈：xuggsx@sina.com 或者 QQ 号 693131033，作者随时解答读者问题。

许国根

本书习题代码下载



说明：本书习题按照“篇”分类，共三个文件夹，请对应正文的习题编号进行查询。

目　录

第 1 篇　关于数据挖掘

第 2 篇　数据挖掘算法

第 4 篇 数据挖掘应用实战

第 1 篇　关于数据挖掘

第 1 章

绪　论

1.1 数据挖掘概述

随着通信、计算机、网络技术和数据库技术的快速发展，以及日常生活自动化技术的普遍应用，如超市 POS 机、自动售货机、信用卡和借记卡、在线购物、自动订单处理、自动售票等，数据正在以空前的速度产生和被收集，而且随着大容量、高速度、低价格的存储设备的相继问世，人们获取数据、存储数据变得越来越容易，数据量急剧增大。在各行各业，许多公司已经认识到信息的重要性，信息即为财富，信息即为竞争优势、信息就是产品正逐渐成为共识……

大量信息在给人们带来方便的同时也带来了大量问题：信息冗余；信息真伪难辨，给信息的正确应用带来困难；网络上的信息安全难以保障；不能搜索到数据中的深层次或隐藏的规律；信息组织形式的不一致，增加了对信息进行有效统一处理的难度等。

缺少如何从海量的数据中提取出有价值知识方法的现状，促使人们产生了对海量数据分析工具的强烈需求。人们期望通过数据分析工具去寻找隐藏在海量数据之后或网络上的更深层次、更重要的信息，理解已有的历史数据并用以预测未来的行为；获得有价值的网络信息和网络服务，为用户提供重要的、未知的信息或知识，指导政府决策、企业决策以获取更大的经济效益和社会效益。为了满足人们对数据分析工具的需求，20 世纪 80 年代后期至今，高级数据分析——基于数据库的知识发现（Knowledge Discovery in Database，KDD）及相应的数据挖掘（Data Mining，DM）理论和技术应运而生。

KDD 是指从数据中发现有用知识的信息和模式的过程，包含数据清理、数据集成、数据选择、数据变换、数据挖掘、模式评价等步骤，最终得到知识。这个过程的输入是数据，输出则是用户期望的有用信息。而 MD 是指使用算法来抽取信息和模式，是 KDD 过程的一个步骤，也是发现中的核心工作。虽然本质上这两者有所不同，事实上在现今的文献中经常把它们等同看待。

数据挖掘可以从技术和商业两个层面上来理解。从技术层面上看，数据挖掘是探查和分析大量数据以发现有意义的模式和规则的过程。从商业层面上看，数据挖掘就是一种商业信息处理技术，其主要特点是对大量业务数据进行抽取、转换、分析和建模处理，从中抽取辅助商业决策的关键性数据。

数据挖掘与传统数据分析方法（如查询、报表、联机应用处理等）有着本质区别：数据挖掘是在没有明确假设的前提下挖掘信息和发现知识。数据挖掘所得到的信息具有先前未知、有效和实用三个特征。先前未知的信息是指该信息是事先未曾预料到的，即数据挖掘是要发现那些不能靠直觉或是经验而发现的信息或知识，甚至是违背直觉的信息或知识。挖掘出的信息越是出乎意料，就可能越有价值。

KDD 过程可以概括为三部分：数据准备、数据挖掘及结果的解释和评估。

数据准备又可分为三个子步骤：数据选取、数据预处理和数据变换。数据选取是指确定目标数据，即根据用户的需要从原始数据库中抽取一组数据。数据预处理一般包括消除噪声、计算补齐缺值数据、消除重复记录、完成数据类型转换等。数据变换是指消减数据维数或降维，即通过一定的方法，减少原始特征或变量的个数（降维），以减少计算工作量。

数据挖掘阶段首先要确定挖掘的任务或目的，即 KDD 要发现的知识类型，如数据分类、聚类、关联规则发现等，然后再确定挖掘算法。在选择算法时既要考虑数据的特点，也要考虑用途或实际运行系统的要求。同样的目标可以选用不同的算法来解决，要做到算法与整个 KDD 过程

的评判标准相一致。

数据挖掘发现的模式，可能存在冗余或无关的模式，或者是不能满足用户的模式，这时需要进行模式的解释和评估，甚至重新开始一个 KDD 过程，以消除冗余或无关的模式，或产生新的模式。有两个影响因素决定数据挖掘过程的质量：一是数据挖掘技术的有效性；二是用于挖掘的数据的质量和数量。错误的数据或不适当的属性，以及数据不适当的转换都不可能发现有效的模式。

可视化技术在数据挖掘的各个阶段都扮演着重要的角色。在数据准备阶段，用户可以使用散点图、直方图等统计可视化技术来显示有关数据，以期对数据有一个初步的了解，从而为更好地选取数据打下基础。在挖掘阶段，用户则要使用一些专业的可视化工具，以显示数据挖掘过程。在表示结果阶段，则要用可视化技术以使发现的知识更易于理解。

在上述步骤中，数据挖掘占据非常重要的地位，它主要是利用某些特定的知识发现算法，在一定的运算效率范围内，从数据中发现有关知识，从而帮助人们在数据库中找到最重要的信息，预测未来的趋势和行为，并做出具有知识驱动的决策，可以说，它决定了整个 KDD 过程的效果与效率。

很显然，数据挖掘有别于传统的数据查询、报表及全文检索等数据分析工作，它常常是在没有前提假设的情况下，从事信息的挖掘与知识的提取。数据挖掘所得到的信息结果，当然不一定全都是先前未知的。

根据数据挖掘的定义，典型的数据挖掘系统具有如下组成部分。

- 数据库、数据仓库或其他信息库：这是一个或一组数据库、数据仓库、电子表格或其他类型的信息库，可以在此数据集上进行数据预处理和选取。
- 数据库或数据仓库服务器：根据用户的数据挖掘请求，数据库或数据仓库服务器负责提取相关数据。
- 知识库：存放领域知识，用于指导搜索或评估结果模式的兴趣度。这种知识可能包括概念分层及用户确信度方面的知识。
- 数据挖掘引擎：数据挖掘的基本组成部分，由一组功能模块组成，用于特征化、关联、分类、聚类分析以及演变或偏差分析。
- 模式评估模块：通常使用兴趣度来测试，并与数据挖掘模块交互，以便将搜索聚集在有趣的模式上。可以使用兴趣度阈值过滤所发现的模式。模式评估模块也可以与挖掘模块集成在一起，其不同在于所用的数据挖掘方法不同。
- 图形用户界面：本模块在用户和数据之间通信，允许用户与系统交互，指定数据挖掘查询或任务，提供信息，帮助搜索聚集，根据数据挖掘的中间结果进行探索式数据挖掘。此外，该模块还允许用户浏览数据库和数据仓库模式或数据结构，评估挖掘的模式，以不同的形式进行模式可视化。

数据挖掘有以下四个特点。

（1）数据挖掘的数据量是非常巨大的，因此，如何高效率地存取数据，如何根据一定应用领域找出数据关系即高效率算法以及是使用全部数据还是使用一部分数据随机或有目的地选择出的数据子集，都成为数据挖掘要考虑的问题。

（2）数据挖掘面临的数据常常是为其他目的而收集好的数据，因此在收集数据时，可能有

一个或几个变量未被收集，但这些变量在后来数据挖掘时被证明是有用的，甚至是至关重要的。也就是说，未知性和不完全性将始终伴随数据挖掘的整个过程。

（3）数据挖掘算法中常常不事先嵌入先验知识。因为这样就等于做“假设检验”。新颖性是衡量一个数据挖掘算法好坏的很重要的原因，当然这些新颖性的结论必须是可以被人们理解的，而不应该是漫无边际的奇怪结论。

（4）数据挖掘中的规则不必适用全部的数据，也不可能挖掘出普遍适用的规则，所有的发现都是相对的，并且只对特定的商业行为具有指导意义。

数据挖掘是一个交叉的学科领域，包括了数据库技术、统计学、机器学习、可视化和信息科学等学科，依赖所挖掘的数据类型或给定的数据挖掘应用，数据挖掘系统也可能集成空间数据分析、信息检索、模式识别、图像分析、信号处理、计算机图形学、Web 技术、数据可视化及经济、商业、生物信息学或心理学等领域的核心技术。数据挖掘中主要采用的技术有人工神经网络、模糊集理论、粗糙集理论、知识表示、归纳逻辑和高性能计算等。通过数据挖掘，可以从数据仓库中提取有趣的知识、规律和信息，并可以从不同的角度观察和浏览。所发现的知识可用于决策、信息管理、查询处理、过程挖掘等。数据挖掘是当今信息技术学科最前沿的领域之一。

1.2 数据挖掘的分类

数据挖掘是一个交叉性的学科领域，涉及统计学原理、模式识别技术、可视化理论和技术等。由于所用的数据挖掘方法的不同，所挖掘的数据类型与知识类型的不同、数据挖掘应用的不同，从而产生了大量的、各种不同类型的数据控制系统。

数据挖掘可根据数据库类型、挖掘对象、挖掘任务、挖掘方法与技术以及应用等方面进行分类。

1. 根据数据库类型分类

此类数据挖掘主要是在关系数据库中挖掘知识。随着数据库类型的不断增加，逐步出现了不同数据库的数据挖掘，如关系数据挖掘、历史数据挖掘、空间数据挖掘、数据仓库的数据挖掘等。

2. 根据数据挖掘对象分类

数据挖掘的对象除数据仓库外，还有多媒体数据挖掘、Web 数据挖掘、文本数据挖掘等。由于对象不同，挖掘的方法有很大的不同，文本、多媒体、Web 等均是非结构化数据，挖掘难度较大。目前 Web 数据挖掘已引起人们的高度关注。

3. 根据数据挖掘任务分类

数据挖掘的任务有关联分析、时序模式、聚类、分类、偏差检测、预测等，所对应的就有关联规则挖掘、序列模式挖掘、聚类数据挖掘、分类数据挖掘、偏差分析挖掘和预测数据挖掘等类型。

数据挖掘还可以按所挖掘知识的粒度或抽象层进行分类，包括采集隐藏于目标数据集中数据的一般性概括知识（高抽象层）的一般性知识挖掘；采集隐藏于原始数据层中的数据的规律性（原始数据层）的原始层知识挖掘；采集多个抽象层上知识的多层知识挖掘等。

4. 根据数据挖掘方法分类

根据所采用的数据分析方法，如面向数据库的方法、面向数据仓库的方法、统计学方法、模式识别方法等，数据挖掘也有不同的分类。如基于概括的数据挖掘，它是利用数据归纳和概括工具，对指定目标数据的一般特征和高层知识进行概括归纳；基于模型的数据挖掘，即根据预测模型挖掘与模型相匹配的数据；基于统计学的数据挖掘，即指针对目标数据，根据统计学原理进行数据挖掘。

5. 根据数据挖掘技术分类

目前，基于数据挖掘技术的分类有自动数据挖掘、证实驱动挖掘、发现驱动挖掘和交互式挖掘等。

（1）自动数据挖掘是指自动地从大量的数据中发现未知的、有用的模式，是数据挖掘的高级阶段。

（2）证实驱动数据挖掘是指用户根据经验创建假设（或模型），然后使用证实驱动操作测试假设（或挖掘与模式匹配的数据），测试的过程即为数据挖掘的过程。所抽取的信息可能是事实或趋势，操作有查询和报告、多维分析和统计分析。其中，查询的目的是有效地表示一个假设，而报告是分析结果的说明。多维分析针对每一维的层次结构，利用特定的查询语句和可视化工具进行分析；统计分析是将统计学与数据挖掘和可视化技术结合进行数据分析。

（3）发现驱动数据挖掘是指在目标数据自动创建一个模型，以预测将来的行为，模型创建的过程即为数据挖掘的过程。所挖掘的知识可能是回归或分类模型、数据库记录间的关系、误差情况等。发现驱动数据挖掘的操作有预测模型化、数据库分割、连接分析（即关联分析）和偏差检测。

近年来，随着人工神经网络和人工智能技术的渗透，发现驱动数据挖掘开始了广泛的应用。

（4）交互式数据挖掘是指利用交互式处理方式，逐渐明确数据挖掘的目标，动态改变数据聚集及搜索方式，逐步加深数据挖掘过程的一种数据方法。

6. 根据数据挖掘应用分类

根据数据挖掘的应用可以将其分成金融数据挖掘、电信数据挖掘、股票市场数据挖掘、WWW数据挖掘等。不同的应用通常需要集成对于该应用特别有效的方法。因此，普通的、全功能的数据挖掘方法并不一定适合特定领域的数据挖掘任务。

1.3　数据挖掘的过程

图 1.1 为数据挖掘的基本过程。但由于数据挖掘的复杂性，往往需要重复以上的某些过程。另外，各过程之间都有直接或间接的关系，不能将它们截然划分。例如数据预处理及变换就包含了线索关系的挖掘。

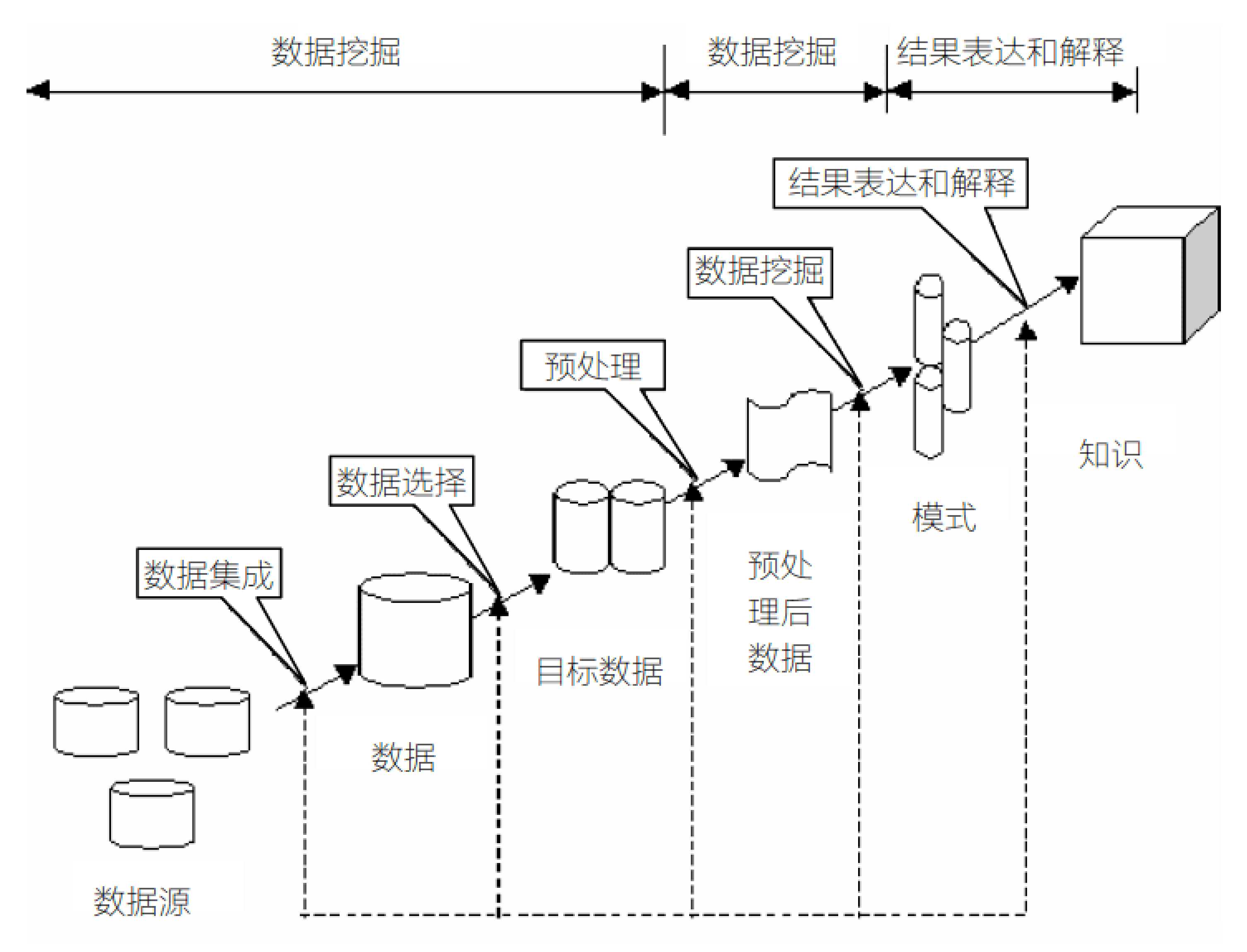


图 1.1 数据挖掘的基本过程

数据准备阶段包括数据收集（集成）和预处理。原始数据的采集部分看似容易且不引人注意，但它却是数据挖掘的基础，要耗用相当大的人力和物力。虽然采用较小规模的数据集也有可能完成数据挖掘，但为了确保挖掘的知识的准确性及预测性，应尽量采集和利用足够多的原始数据。

在数据采集之后，就需要对数据进行选取和预处理。数据选取就是在原始数据中，将有代表性的数据提取出来组成样本。预处理就是将一些不完全数据、噪声数据以及矛盾数据等不适合用来训练和学习的数据排除在样本集外。当数据结构较为简单且合理、数据较为齐全时，可以直接利用原始数据进行数据挖掘，但挖掘输出结果质量并不能保证。在进行数据挖掘前对数据进行必要的“整理（预处理）”和“筛选（选取）”，能够提高数据挖掘的效率与准确性。

数据转换是将不符合数据挖掘算法要求格式的数据转换成一定格式，或对数据维数进行降维。转换完成后，如果对数据样本集不满意，就应该返回到上一阶段，重新对原始数据进行选取和预处理工作，反之，进行下一步。

数据仓库是一种数据存储的有效形式，其非常有利于数据挖掘。它利用信息技术所提供的海量数据存储、分析能力，将数据经过整理、规划而建立成一个强大的数据管理智能系统，可以协助数据挖掘以及决策的进行。

数据仓库建立之后，就可以使用各种数据挖掘的算法。首先根据特定的问题领域的性质，选择有明显区分意义的特征，这常常是数据挖掘过程中非常关键的一步。合适的特征向量以及维数，能保证数据挖掘过程的有效性和准确性，不会浪费计算时间及产生过拟合问题。完整的数据样本有利于选择特征，当然也可以利用先验知识补齐缺少的或删除不合理的数据。例如分类问题中，在选择或设计特征向量时，应选择那些容易获得、对不相关变换保持不变、对噪声不敏感、能较易区分不同类别模式的特征集。这不仅是一个技术问题，也是一个经验问题。然后再选择合适的模型或具体算法。在设计算法和数据结构时，一定要考虑效率问题，以及如何及时更新所提出的算法以适应数据库的变化。要根据问题领域和数据的结构、算法的特点以及算法的计算资源消耗与计算复杂度等因素选择较为合理的挖掘算法。一个理想的算法不仅能得到准确的知识，而且对计算时间及存储容量等硬件性能的要求要低。例如某些问题中，在不考虑工程上约束的前提下，

确定能够设计一个性能非常优秀的分类器，但是如果存在工程上的约束，就不一定能够得到同样性能的分类器。同时还应记住的是没有一种通用方法可以解决所有的问题。一般认为，反复试验和基于样本的方法是设计模型最有效的方法。

通过数据挖掘算法，就可以得到隐藏在数据中的知识，并用此对已往的数据进行验证，并对发展趋势进行预测。如果验证或预测结果不理想，说明没有得到所需要的知识，则需要返回到上一阶段，甚至重头来过，重新执行上述过程。

1.4　数据挖掘的任务

数据挖掘的任务有如下 7 类。

1. 概念描述

概念描述本质上就是对某类对象的内涵特征进行概括。概念描述分为特征化描述和区别性描述。前者是指描述目标类数据的一般特征和特性的汇总；后者是将目标类对象的一般特性与对比类对象的特性进行比较。

2. 关联分析

关联分析就是发现数据特征间的相互依赖关系，通常是在给定的数据集中发现频繁出现的模式知识（又称为关联规则）。若两个或多个数据项的取值之间重复出现且概率较高时，就存在某种关联（或依赖关系），即从一个元素 A 的值可以推出另一个元素 B 的值，这里的元素可以是字段，也可以是字段间的关系。这样就可以建立起这些数据项的关联规则。例如买面包的顾客有 90%的人还会买牛奶，这就是一条关联规则。若根据这条规则，在商场中将面包和牛奶放在一起销售，可以提高它们的销量。但要注意的是关联规则并不是因果关系，它不代表实际数据或现实世界中的内在因果关系。

在大型的数据库中，这种关联规则很多，需要进行筛选。一般用“支持度”和“可信度”两个阈值来淘汰那些无用的关联规则。

- “支持度”表示该规则所代表的事例（元组）占全部事例（元组）的百分比；
- “可信度”表示该规则所代表事例占满足前提条件事例的百分比。

3. 时间序列分析

在时间序列分析中，数据的属性值是随着时间不断变化的，并且在一般情况下，时间间隔是相等的。

时间序列分析有三个基本功能：第一，使用距离度量来确定不同时间序列的相似性；第二，通过检验时间序列图中线的结构来确定时间序列的行为；第三，利用历史时间序列预测数据的未来数值。

4. 分类分析

分类分析是数据挖掘中一项非常重要的任务，它是利用已知数据库元组和类别的训练样本集

通过相关算法找出一个类别的概念描述，即该类的内涵描述，它代表了该类别的整体信息，一般用规则或决策树模式表示，该模式把数据库中的数据项映射到给定类别中的某一个。分类分析已广泛应用于用户行为分析（受众分析）、风险分析、生物科学等领域。

类的内涵描述可分为特征描述和辨别性描述。特征描述是对类中对象的共同特征的描述；辨别性描述是对两个或多个类之间的区别的描述。特征描述中允许不同类具有共同特征，而辨别性描述中不同类别不能有相同的特征。辨别性描述用得更多。

目前，分类方法的研究成果较多。可从三个方面判别分类方法的好坏：第一，预测准确度，即对非训练样本集数据的判别准确度。第二，计算复杂度，即计算分类时时间和空间的复杂度。第三，模式的简洁度，在同样的情况下，希望决策树小或规则少。

5. 聚类分析

聚类分析试图找出数据集中数据的共性和差异，并将具有共性的对象聚合在相应的簇中。聚类分析可以帮助判断哪些组合更有意义，已广泛应用于客户细分、定向营销、信息检索等领域。

聚类分析与分类分析不同。在聚类过程中，需要划分的类是未知的。通过确定数据库中的数据之间在预先指定的属性上的相似性，就可以将它们划分为一系列有意义的子集，即类。在同一类别中，个体之间的距离较小，而不同类别的个体之间的距离偏大。聚类增强了人们对现实世界的认识，即“物以类聚”。

聚类方法包括统计分析方法、机器学习、神经网络方法和面向数据库的方法。在统计学的研究中，主要集中于基于距离的聚类分析，在这里距离是指欧氏距离、马氏距离等。在机器学习领域，聚类是无监督的学习，是观察式学习，在这里距离是根据概念的描述来确定的，故也称概念聚类。当聚类对象动态增加时，概念聚类则称为概念形成。在数据挖掘中的聚类研究主要集中在大型数据库中的聚类分析方法的构成。

6. 离群点检测

数据库中的数据存在很多异常情况，从数据分析中发现这些异常情况也是非常重要的，它也称孤立点分析。异常包括以下几种模式：不满足常规类的异常例子、出现在其他模式边缘的奇异点、在不同时刻发生了显著变化的某个元素或集合、观察值与模型推测出的期望值之间有显著差异的事例等，其基本思想是寻找观察结果与参照量之间的有意义的判别。参照是给定模型的预测、外界提供的标准或另一个观察。离群点检测已广泛应用于（商业、金融、保险等领域）欺诈行为的检测、网络入侵检测、反洗钱、犯罪嫌疑人调查、海关、税务稽查等领域。

7. 预测

预测是利用历史数据找出变化规律，建立模型，并用此模型来预测未来数据的种类、特征等。

典型的预测方法有回归分析、时间序列分析以及神经网络分析等。回归分析是利用大量的历史数据，以时间为变量建立线性或非线性回归方程。预测时，只要输入任意的时间值，通过回归方程就可求出时间的状态。时间序列分析是在分析序列结构特点的基础上，利用参数模型等方法以过去的数据来判定一个变量的未来趋势及不同变量间同期或前后期的关联性。神经网

络方法能实现非线性样本学习，能进行非线性函数的判别，既可以用于连续数值也可以用于离散数值的预测。

1.5　数据挖掘的对象

数据挖掘的对象原则上可以是各种存储的信息。目前的信息存储方式主要包括关系数据库、数据仓库、事务数据库、高级数据库系统等的各种数据库、文本数据、图像、视频数据和 Web 数据。其中，高级数据库系统包括面向对象数据库、关系对象数据库以及面向应用的数据库（如空间数据库、时态数据库、文本数据库、多媒体数据库等）。

1.5.1　数据库

1. 关系数据库

关系数据库由表组成，每个表有一个唯一的表名。属性（列或域）集合组成表结构，表中数据按行存放，每一行称为一个记录。记录间通过键值加以区别。关系表中的一些属性域描述了表间的关系，这种语义模型就是实体关系模型。关系数据库是目前最流行、最常用的数据库之一，为数据挖掘研究工作提供了丰富的数据源。

2. 数据仓库

数据仓库是一种管理技术。根据数据仓库系统构造方面的设计师 W.H.Inmon 对数据仓库的定义，数据仓库就是面向主题的、集成的、非易失性的、随时间变化的数据集合，用以支持管理人员的决策。通常构造数据仓库是将多个异种数据源（如关系数据库、一般文件和联机事务处理记录）集成在一起。使用数据清理和数据集成技术，确保命名约定、编码结构、属性度量等的一致性。非易失性是指数据仓库反映的是历史数据的内容，而不是日常事务处理产生的数据，数据经加工和集成进入数据仓库后是很少修改或根本不修改的，供管理人员决策分析使用。随时间变化指数据仓库是不同时间的数据集合，它要求数据仓库中的数据保存时限以满足进行决策分析的需要，而且数据仓库中的数据都要标明该数据的历史时期。

数据仓库根据多维数据库结构建模，每一维代表一个属性集，每个单元存在一个属性值，并提供多维数据视图，允许通过预计计算快速地对数据进行总结。尽管数据仓库中集成了很多数据分析工具，但仍然需要像数据挖掘等更深层次、自动的数据分析工具。

需要注意的是数据仓库不同于数据库。数据仓库是一种解决方案，是对原始的操作数据进行各种处理并转换成有用信息，用户可以通过分析这些信息做出策略性决策。因此，在很多场合，数据仓库也称为“决策支持系统”。

3. 事务数据库

一个事务数据库由文件、每条记录代表一个事务。典型的事务包含唯一的事务标识，多个项目组成一个事务。事务数据库可以用额外附加的关联表来记录其他信息，比如销售方面的事务交易日期、顾客 ID 及交易发生的部门等信息。

4. 面向对象数据库

面向对象数据库是基于对象程序设计的范例，其每一个实体作为一个对象。与对象相关的程序和数据封装在一个单元中，通常用一组变量描述对象，等价于实体关系模型和关系模型中的属性。对象通过消息与其他对象或数据库系统进行通信。对象机制提供一个模式获取消息并做出反应的手段。类是对象共享特征的抽象。对象是类的实例，也是基本运行实体，可以把对象按级别分为类和子类，实现对象间属性共享。

5. 关系对象数据库

关系对象数据库的构成基于关系对象模型。为操作复杂的对象，该模型通过提供丰富数据类型的方法进一步扩展了关系模型。在关系查询语言中增加了新增类型的检索能力。关系对象数据库在工业和其他应用领域使用越来越普遍。与关系数据库中的数据挖掘相比，关系对象数据库中的数据挖掘更强调操作复杂的对象结构和复杂数据类型。

6. 空间数据库

空间数据库包含空间关系信息，有地理（地图）数据库、医学图像数据库和卫星图像数据库等类型。空间数据可以用 n 维位图、像素图等光栅格式表示，也可以用向量形式表示（例如道路、桥梁、建筑物等基本地理结构可以用点、线、多边形等几何图形表示为向量形式）。空间数据库中的数据挖掘可以提示地理数据中某种类型区域中的建筑物特征，也可以揭示医学图像数据库的图像信息与对应的疾病间的关系。

7. 时态数据库和时间序列数据库

时态数据库和时间序列数据库均存储与时间有关的信息。前者通常存储与时间属性相关的数据，这些属性可以是具有不同语义的时间戳；而后者存储的是随时间顺序变化的数据。数据挖掘技术可以用于发现对象演变特性或数据库中数据的变化趋势，时间可以是财政年、教学年、日历年等，也可以是年细分的季度或月。

数据库具有以下特点。

- 数据动态性：数据的动态变化是数据库的一个主要特点。由于数据的存取和修改，使数据的内容经常发生变化，这就要求数据挖掘方法能适应这种变化。
- 数据不完整性：数据的不完整性主要反映在数据库中记录的域值丢失或不存在（空值）。这种不完整性数据给数据挖掘带来了困难。为此必须对数据进行预处理、填补该数据域的可能值。
- 数据噪声：由于数据录入等原因，造成错误的数据，即数据噪声。含噪声的数据挖掘会影响抽取模式的准确性，并增加了数据挖掘的难度，可以用概率的方法消除噪声。
- 数据冗余性：数据冗余性表现为同一信息在多处重复出现。函数依赖是一个通常的冗余形式。冗余信息可能造成错误的数据挖掘，至少有些挖掘的知识是用户不感兴趣的。为避免这种情况的发生，数据挖掘时，需要知道数据库中有哪些固有的依赖关系。
- 数据稀疏性：数据稀疏性表现为实例空间中数据稀疏，数据稀疏会使数据挖掘丢失有

用的模式。

- 海量数据：由于数据库中的数据不断增长，已出现很多海量数据库。数据挖掘方法需要逐步适应这种海量数据挖掘，如建立有效的索引机制和快速查询方法等。

1.5.2 文本

文本是用文字串描述对象的数据文件。这里的文字不是通常所说的简单的关键字，可能是长句子或图形，比如产品说明书、出错或调试报告、警告信息、简报等文档信息。

文本分析包括以下几类。

- 关键词或特征提取：一篇文本中，标题是该文本的高度概括。标题中的关键词是标题的核心内容。关键词的提取对于掌握该文本的内容至关重要。文本中的特征如人名、地名、组织名等是某些文本中的主体信息，特征提取对掌握该文本的内容很重要。
- 相似检索：文本中的关键词的相似检索是了解该文本内容的一种重要方法。例如“专家系统”与“人工智能”两个关键词是有一定联系的，研究专家系统的文本一定属于人工智能的研究领域。
- 文本聚类：对于文本标题中关键词（主题词）的相似匹配是对文本聚类的一种简单方法。定义关键词的相似度，将便利文本的简单聚类，类中文本满足关键词的相似度，类间文本的关键词超过相似度。
- 文本分类：将文本分类到各文本类中，一般需要采用一个算法。这些算法包括分类算法、近邻算法等。这需要按文本中的关键讯号或特征的相似度来区分。

1.5.3 图像与视频数据

图像、音频、视频等信息存储在多媒体数据库中。多媒体数据库管理系统提供在多媒体数据库中对多媒体数据进行存储、操纵和检索的功能，特别强调多种数据类型间（如图像、声音）的同步和实时处理，主要应用在基于图片内容的检索、语音邮件系统、视频点播系统。多媒体数据挖掘、存储和检索技术需要集成标准的数据挖掘方法，还要构建多媒体数据立方体，运用基于模式相似匹配的理论等。

图像与视频的数据挖掘包括以下几类。

- 图像与视频特征提取：图像与视频数据特征有颜色、纹理和形状等。这些特征提取用于基于内容的相似检索。
- 基于内容的相似检索：根据图像、视频特征的分布、比例等进行基于内容的相似检索，可以将图像和视频数据进行聚类以及分类，也能完成对新图像或视频的识别。如对遥感图像或视频的识别可以应用于森林火灾的发现与报警、河流水灾的预报等。
- 视频镜头的编辑与组织：镜头代表一段连续动作（视频数据流）。典型的镜头或足球赛的射门、某段新闻节目等，需要在冗长的视频数据流中进行自动截取。

经过编辑的镜头，按某种需要重新组织，将形成特定需求的新视频节目，如足球射门集锦、某个新闻事件的连续报道等。

1.5.4 Web 数据

自数据挖掘技术在 20 世纪末兴起以来，它的挖掘对象已经发生了很多变化。近几年，基于互联网应用的数据挖掘开始发展以来，其挖掘对象往往是半结构化的、异构的数据（Web 数据）。互联网挖掘的核心是数据挖掘和 KDD 技术在互联网相关的数据源上的延伸，它是面向互联网数据进行分析和知识提取的。互联网中页面内部、页面间、页面链接、页面访问等都包含大量对用户有用的信息，而这些信息的深层次含义是很难被用户直接使用的，必须经过浓缩和提炼。从某种意义上讲，这正是互联网挖掘所解决问题的出发点和目标。

Web 数据挖掘具有以下两个特点。

1. 异构数据集成和挖掘

Web 上每一个站点是一个数据源，各数据源都是异构的，形成了一个巨大的异构数据库环境。将这些站点的异构数据进行集成，给用户提供一个统一的视图，才能在 Web 上进行数据挖掘。

2. 半结构化数据模型抽取

Web 上的数据非常复杂，没有特定的模型描述。虽然每个站点上的数据是结构化的，但各自的设计对整个网络是一个非完全结构化的数据，称为半结构化数据。对半结构化数据模型的查询和集成，需要寻找一个半结构化模型抽取技术来自动抽取各站点的数据。

互联网挖掘可以分为互联网结构挖掘、互联网元数据挖掘、互联网使用挖掘、互联网内容挖掘、总结和摘要系统 5 种主要任务。

（1）互联网结构挖掘：互联网结构挖掘是对互联网页面之间的链接结构进行挖掘。

（2）互联网元数据挖掘：元数据就是指那些能够帮助识别、描述和定位互联网资源的数据。因为元数据能够在很大程度上反映 Web 文档的特征，所以元数据挖掘可以提高互联网知识挖掘的准确性。

（3）互联网使用挖掘：互联网使用挖掘是对用户访问互联网时在服务器留下的访问记录进行挖掘，即对用户访问互联网站点的存取方式进行挖掘。挖掘的对象是在服务器上的包括 server log data 等在内的日志文件记录。

（4）互联网内容挖掘：互联网内容挖掘是指对站点的互联网页面内容进行挖掘。

（5）总结和摘要系统：目前，互联网上的信息量正以爆炸式的方式增长，它已经远远超过人类的阅读能力。互联网上信息的总结和摘要系统正是应这种实践要求产生的。它通过各种信息抽取方法，希望把互联网蕴藏的信息抽取出来，将信息浓缩或升华，然后，或者形成文字，或者使用数值的方法，例如信息流来表示信息浓缩或升华的内容。

1.6 数据挖掘建模方法

一个成功的数据挖掘并不是对数据的简单运用，而是要在大量数据中不仅发现潜在的模式，而且必须能对这些模式做出反应，对它们进行处理，将数据转化为信息，将信息转化为行动，最终将行动转化为价值。所以为了成功运用数据挖掘，对数据挖掘技术层次的理解至关重要，尤其

是应该了解如何将数据变成有用信息的过程。

1999 年欧盟机构联合起草了 CRISP－DM，目前在各种 FDD 过程模型中得到广泛的应用。它强调 DM 不单是数据的组织或者呈现，也不仅是数据的统计建模，而是一个从理解业务、寻求解决方案到接受实践检验的完整过程，图 1.2 即为一个 DM 完整过程的描述，它可分为业务理解、数据理解、数据准备、建模、评估和部署 6 个阶段。

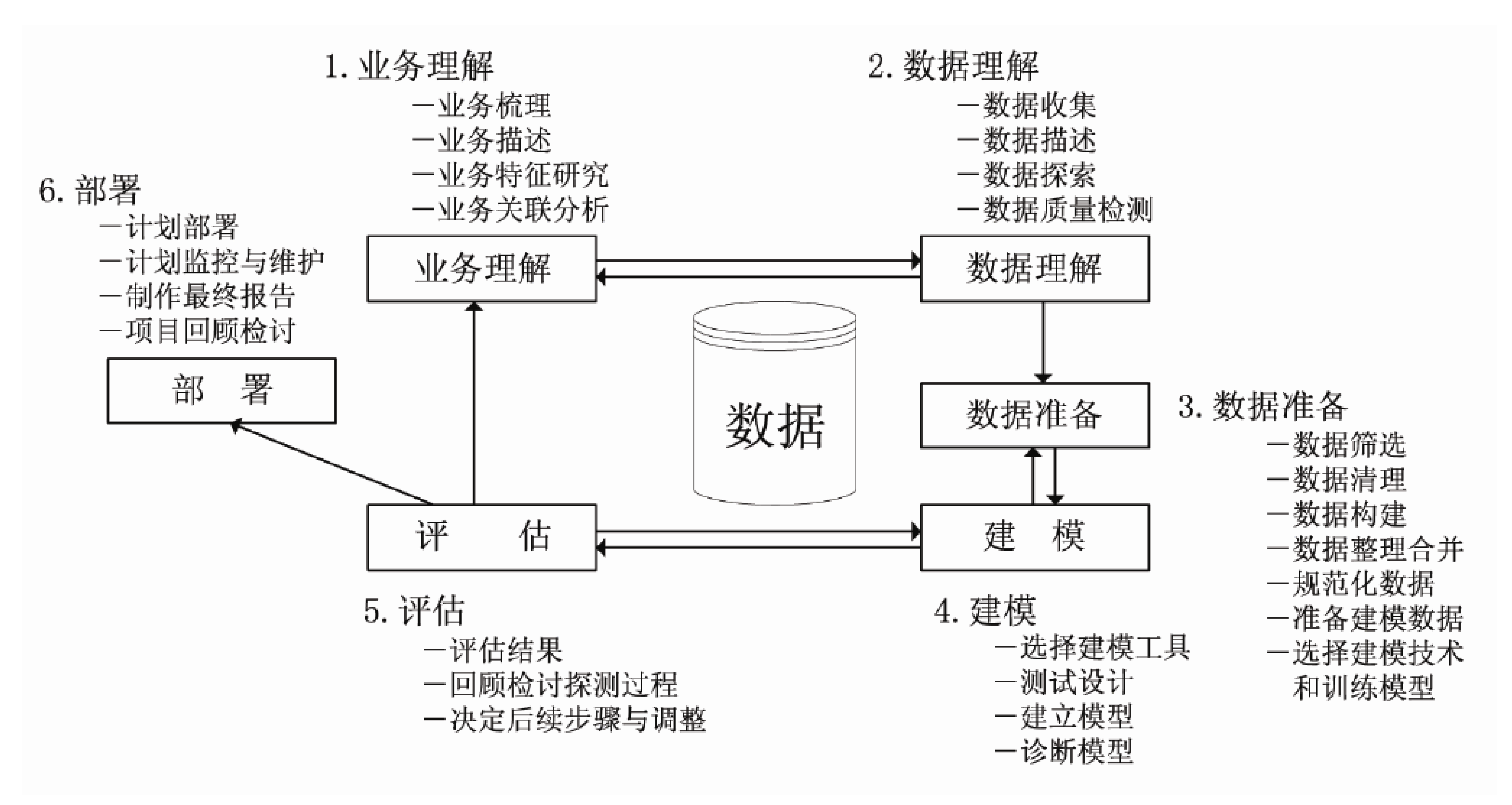


图 1.2　CRISP－DM 处理流程

数据挖掘过程是一个人机交互、多次反复的过程，CRISP－DM 处理流程的 6 个阶段的顺序并不是固定的，通常需要在不同阶段之间来回以逐步完善。在实际应用中，应该针对不同的应用环境和实际情况做出必要的调整，使数据挖掘根植于业务环节中。一个数据挖掘项目通常并不是一次性执行 6 个步骤就结束了，往往需要反复迭代、不断完善。从一个数据挖掘循环获得的知识通常会产生新的问题，出现新的机会来识别和满足客户的需求。通常可以在新一轮的数据挖掘过程中找到解决这些问题的方法，并把握新的机会来满足客户更高的需求。

1.6.1　业务理解

业务理解是数据挖掘过程中的第一个阶段，主要集中在理解项目的目标和从业务的角度理解需求，同时将这个需求转化为数据挖掘问题的定义和完成目标的初步计划。具体而言有业务梳理、业务描述、业务特征研究和业务关联分析等过程。通过业务理解可以明确是否需要进行数据的挖掘。

业务梳理和业务描述需要进行不断的探索、交流，从而正确理解问题。这就要求数据挖掘技术人员不仅需要充分了解技术和数据，还必须和了解企业业务问题的人员（行业专家）沟通与交流，以明确他们的业务问题。此外，在分析的最后阶段，只有行业专家才有资格判断最后结果的优劣。

在业务特征研究阶段，需要确定诸如哪类客户有可能对产品感兴趣、客户具哪些基本特征、每位客户能创造多少价值、能创造较高价值的客户应具有的共同特征等各种目标，并且对业务进行关联分析，寻找业务间的隐含的关联。

在数据挖掘实践中，数据挖掘技术与业务的需求结合起来是最困难也是最重要的一点。数据

挖掘专家与业务专家需要相互学习一些对方的专业知识，以便两者都能互换角度，能够知道现有技术在这个业务中能做些什么，才能共同确定数据挖掘项目的目标，才能使数据挖掘技术发挥出应有的效益和效率来。

1.6.2 数据理解

数据理解从数据收集开始。在收集数据前，需要明确所需的信息，然后根据相应的标准收集必需的数据。通过对所分析业务的了解，掌握数据来源的分析和了解，及对先验知识的收集、学习、整理。这一步工作处理得好，就为数据挖掘奠定了一个良好的基础。

数据收集后，就需要了解数据的关系和质量，包括数据的整体模型、数据之间的关系、数据的质量等，以发现数据的内部属性，或是探测引起兴趣的子集去形成隐含信息的假设。在这个过程中，需要应用一些数学工具对数据进行描述，理解数据的内涵与数据的分布特征，检验数据的"总和"或者"表面的"特征，并检验数据的质量，确定数据是否完整、正确，是否存在缺失值，变量的含义与变量值是否一致等质量问题。在此基础上，详细分析数据的变量特征，识别潜在的特征，思考和评估在描述数据过程中的信息和发现，提出假设并确定方案，阐明数据挖掘的目标。

1.6.3 数据准备

数据准备包括从原始数据中创建目标数据集。它有以下 6 个任务。

（1）数据筛选：数据筛选是确定数据挖掘分析过程中所必需的数据，即选择有用的特征和记录。在选择数据时，首先要考虑数据应符合问题的需要，并且采集尽可能多的数据量，尤其在使用抽样调查数据时，应注意数据的普遍性。有时还需要收集期望的输出等。

数据挖掘是基于对海量数据的分析的，但在实践应用中既要考虑数据挖掘的结果，又要考虑数据挖掘的成本，很多时候，需要从超大的数据集中选取有代表性的数据进行分析。

（2）数据清理：数据清理是清理数据中包含的噪声和与数据挖掘主题明显无关的数据。它通常包括填补空缺的数据值、清理噪声数据以及解决数据不一致的问题。

（3）数据构建：数据构建是指属性构造，多维数据组织（聚集）和数据泛化处理等任务。

属性构造是指通过组合、汇总、提取等方式在已有属性的基础上构造新的属性，以帮助提高数据挖掘的质量。多维数据组织是指对数据进行汇总和聚集，采用切片、放置等操作使原始数据按照多维立体形式成为不同层次、不同粒度和不同维度的聚焦。而数据泛化则是指使用高层次的概念替换低层次的概念。

（4）数据整合：数据整合是将来自多个数据源的相关数据组合在一起，即把不同来源、格式、特点的数据在逻辑上或者物理上有机地集合在一起，使之更加有利于数据挖掘过程的实现。

（5）数据规范化：数据规范化就是将数据的属性数据按比例缩放，使之落入一个特定的小范围内，以消除数值型因大小不一致或度量单位的不同而造成的数据挖掘结果出现偏差。

（6）准备建模数据集：对数据进行上述处理后，并不能直接用于数据建模，还需要考虑到数据的稀疏程度。通常对于稀疏的数据，最好选用 15%～30%的比例来建模。

为了评估模型，一般将建模数据分成三个部分，即训练集、测试集和评估集。将数据的训练集最初用于建立模型的，用测试集和评估集来精化模型和评估模型。

在实践中，一般保留 1/3 的数据用于测试，2/3 的数据用于训练，同时在随机取样时必须确

保在训练集和测试集中每个类各自应有的比例（分层）。更通用的方法是用不同的随机样本重复进行多次训练和测试，每次迭代过程中，随机抽取一个特定比例的数据进行训练，即交叉验证。首先确定一个固定的比例或折数（如 3 折），将数据集分成三部分，每部分轮流用于测试而剩余的则用于训练。重复此过程 3 次，从而每个实例恰好有一次是用于测试的。给定一个数据样本，预测某种机器学习技术误差率的标准方法就是使用分层 10 折。数据被随机分割成 10 个部分，每部分中的种类比例与整个数据集中的比例基本一致，每部分依次轮流被用于测试，而其余 9/10 则参与某一个学习算法的训练。

1.6.4　建模

数据挖掘中的建模是指根据问题的特定对象而建立模型，并利用模型解决特定的现象和预测对象的未来。为此，建模时应注意以下 3 点。

1. 预测模型的时间范围

在建立模型的过程中，首先需要训练模型，即用历史数据构建模型进行预测，然后将模型应用于新的数据中从而生成结果（得分）。在这个过程中，需要关注训练模型的时间间隔和模型产生得分这两个时间范围。训练过程中产生的结果是已知的，得分过程所产生的结果是未知的。模型建立后，它的执行效果只能通过已知的历史数据来评估，在有些情况下，用历史数据得到的是好结果，但用在预测结果却不理想。因此为了更加有效地对未来问题进行预测，不仅需要了解构建模型的过程，还要了解模型的工作情况。

2. 模型的使用有效期

在建立模型时，还要考虑模型的使用有效期问题，即模型使用有效期和模型预测有效期。前者是指在业务环境、技术手段、客户基础等相对稳定的条件下，可以使用模型的时间期限。一旦条件改变，就要用新的数据构造新的模型。后者是指预测结果应该在特定的时间内才有效。例如用电高峰与低谷的模型很明显是不一致的。

3. 建立模型的假设

模型的成功应用依赖于 3 个基本的假设：一是历史是未来的写照；二是数据是可以获得的；三是数据中包含预期目标。

以上 3 个基本假设都是在一定的条件下才能存在。对于第一个假设，要求待解决的问题和客户的环境要前后一致；对于第二个假设，要求数据可以通过一定的技术手段获得，数据中不能有太多的缺失值或格式有错误等问题；对于第三个假设，则要求预测目标不能发生改变。

下面讲解如何建立有效的模型。

建立模型最重要的目标是保持模型的稳定，即要求在使用模型进行预测时，必须保证未来预测值也是正确的。为此，建立一个有效的模型需考虑以下几个问题。

（1）数据收集要充分，这样才能保证训练集、测试集和评估集 3 个子集都有一定的数量。

（2）对于类别不平衡的数据，通过抽样来控制模型集的密度，即不同分布的类别比例。

（3）注意数据的输入和输出时间范围。

（4）在模型集中使用多重窗口有助于确保模型稳定，并且在时间上易于转换。

（5）大多数建模过程需要建立多个模型，并对多个模型的效果进行比较，以选用效果最好的模型进行预测，或者对多个模型进行组合，以得到性能更优的集成分类模型。

（6）对不同的模型集、模型参数等进行试验，有助于建立更好更稳定的模型。

数据挖掘是一项具有挑战性、探索性和需求不断创新的学科与技术。在建模过程中，不能被限制和约束，既要重视经典的有关数据挖掘的原理，但可能还可借鉴更多的学科知识，开创数据挖掘新的原理和新的理论，同时，要注意总结经验，通过不断增长的各种数据挖掘的业务需求来寻找和探索新理论与新知识。新思想、新技术往往来源于其他领域。

1.6.5 评估

评估是将模型的输出结果与现实生活中发生的结果进行比较，从而进一步评估模型。为了保证预测结果的有效性，对模型进行评估时应遵循以下原则。

（1）合理性：模型应具有与事物的发展规律相一致的特性，且符合逻辑。

（2）预测能力：模型的预测能力表现在两个方面：一是模型能否说明所要预测期间事物的发展情况；二是预测的误差，即只有预测结果有一个合适的置信区间，才能保证预测有意义。

（3）稳定性：模型的稳定性是指模型能在较长的时间内准确地反映预测的发展变化情况，以及其参数和预测能力受统计数据变化影响的程度。如一个模型无论是用 2000 年的数据还是用 2005 年的数据建立起来的，其参数和预测能力变化不大，说明此模型较为稳定。

（4）简单性：当两个模型的预测能力相差不大时，形式简单、容易运用的模型是优先选择的对象。

（5）成本要低。即当模型发生错误的预测时，所造成的损失要小。

评估结束后，需要对整个数据挖掘过程进行回顾，查找及分析预测误差的大小及原因，以决定后续的数据挖掘的步骤并做出相应的调整。

评估是数据挖掘能否取得成功的关键一环。在训练集上表现好的算法并不意味着在独立的测试集中或实际数据中也会有好的结果。

为了评估模型效果，数据挖掘过程中所使用的 3 个数据集必须保持独立性，测试数据不能以任何方式参与模型的建立，而对模型进行参数的优化，也必须使用不同于模型建立所使用的数据。验证数据必须有别于训练集以获得较好的优化或选择阶段的性能，同时测试数据集也必须有别于其他两个数据集以获得对真实误差的可靠估计。但如果知道了模型的误差率，便可以将测试数据合并到训练数据中；同样，一旦验证数据已被使用，那么也可以将验证数据合并到训练数据中，使用尽可能多的数据重新训练模型。

为得到可靠的结果，一般使用 10 次 10 折交叉验证。当然也可以用其他验证方法，如留一交叉验证法。此法实际上就是 n 折交叉验证法，其中是数据集中所含实例的个数。每个实例依次被保留而用于测试，其余的数据则用于训练。由于尽可能用了最多的数据参与训练，从而可能会得到更准确的模型。当然它的计算量也相应增大了。

还有一种评估方法是基于统计学的放回抽样过程。也即一个有 n 个数据集进行了 n 次放回抽样，从而形成了另一个拥有 n 个数据的数据集。这个新的数据集必有重复的数据，原始数据集中也必有部分数据未被抽样，这些数据就可以用作测试数据。

1.6.6　部署

模型的作用是从数据中找到知识，获得的知识需要以便于用户使用的方式重组和展现。所以模型的建立并不是项目的结束，在模型建立并验证后，一般由用户把模型预测的结果作为参考，提出解决业务问题的方案，从而做出部署。

根据需求，这个阶段可以产生简单的报告，或者是实现一个比较复杂的、可重复的数据挖掘过程，其任务包括：计划部署、计划监控与维护、制作最终报告项目回顾和总结。

1.7　数据挖掘的应用

数据挖掘技术来源于商业的直接需求，并在各种领域都有广泛的使用价值。数据挖掘已在银行、金融、零售、医药、电子工程、航空、旅馆等行业具有大量数据和深度分析需求的、易产生大量数字信息的领域得到广泛的使用，并带来了巨大的社会效益和经济效益。数据挖掘技术既可以检验行业内长期形成的知识模式，也能够发现隐藏的新规律。随着更多行业数据挖掘的应用成功，数据挖掘的应用前景十分广阔。

1.7.1　在金融领域中的应用

在金融方面，银行和金融机构往往持有大量的关于客户的、各种服务的以及交易事务的数据，并且这些数据通常比较完整、可靠和高质量，这极大地方便了系统的数据分析和数据挖掘。在银行业中，数据挖掘被用来建模、预测、识别伪造信用卡、估计风险、进行趋势分析、效益分析、顾客分析等。在此领域运用数据挖掘，可以进行贷款偿付预测和客户信用政策分析，以调整贷款发放政策，降低经营风险。

信用卡公司可以应用数据挖掘中的关联规则来识别欺诈。股票交易所和银行也有这方面的需求。对目标客户群进行分类及聚类，以识别不同的客户群，为不同的客户提供更好的服务，以推动市场。

数据分析工具可以找出金融交易的异常模式，以侦破洗黑钱和其他金融犯罪活动。洗钱是一种非理性的经济活动，因而必然表现出不同于正常理性的经济活动特征。通过研究离群点（交易金额异常增大、近似等额、交易频率的异常变化）检测以及关联分析（如账户日常交易的信息如账号、交易时间、交易名称、公司名称、企业行业代码、企业性质、企业的信用等级、注册资金等）就可以识别可疑洗钱的行为模式，从而准确、及时地对各种信用风险进行监视、评价、预警和管理，评价这些风险的严重性、发生的可能性以及控制这些风险的成本，进而可以采取有效的规避和监督措施，从而可以在信用风险发生之前对其进行预警和控制，趋利避害，做好防范工作。

数据挖掘技术在我国金融业的应用正处于起步阶段。我国金融业对信息化工作的重视在近些年达到了前所未有的高度，特别是数据大集中工程的实施，使得我国金融业的硬件建设方面与国际发展的步调基本一致，但同时也提出一个新课题，即如何处理每日在互联网上产生的海量数据，进行科学的分析处理，并及时提供决策支持。数据挖掘可以在这方面起到非常关键的作用。

数据挖掘还可以在股票市场发挥重要的作用。股票交易的时序数据是一种常见的数据结构，对股市进行动态数据挖掘，可以随时掌握由大量数据所反映的金融市场暗流。通过过滤股市的各种交易数据，找出非法的炒作现象和操作，例如通过对异常交易数据的分析，判断是否存在非法

交易。并且还可以将监管搜索范围扩大到一般的网页上，以适应网上股民数量日益增多的特点，并借助一定的文字分析技术提高准确率，这对稳定我国的金融市场有着积极作用。

数据挖掘在股市的另一个应用是研究股市炒作的快速监测算法和技术。我国的股市都是电子交易，这些交易每天产生的海量数据已超出人工处理的能力，但这正使得应用计算机算法进行智能自动监测成为可能。从管理部门角度出发，可以通过过滤股市的各种交易数据发现异常现象和相应的操作，识别出合法和非法的炒作，找出中国式的股市各种炒作的模式。

数据挖掘在打造金融行业知识创新型企业中，正在发挥着重要作用。信息的分析与管理，在整个信息的获取和信息的使用之间，搭建了一个有效的渠道，通过对于海量源的数据的抽取、转化和加载，向金融企业用户提供统计报表、多维分析、决策支持等相关的信息和知识。目前大规模海量的数据库挖掘已成为数据挖掘研究的主流之一。

1.7.2 在零售业中的应用

在零售业方面，计算机使用率已经越来越高，大型的超市大多配备了完善的计算机及数据库系统。随着条形码技术的广泛使用，目前我国大部分商业零售企业已经基本配备了销售点（point of sales，POS）系统，部分商场甚至配备了决策支持系统和库存管理系统。随着交易的持续进行，记录了大量的客户交易以及销售、货物进出与服务记录等大量数据。同时超市行业的迅速扩张，经营规模的不断扩大及竞争的日趋激烈，使它们对采购管理技术、商品配送技术、信息技术和整体营销技术提出了新的要求。这些需求使得数据挖掘技术在零售行业大有用武之地。利用数据挖掘技术，零售企业可以更好地掌握客户信息，及时地识别顾客购买模式和趋势，发现潜在的购买需要，从而通过改进服务质量，大大减少优惠促销方式的盲目性，取得更高的顾客保持力和满意程度，减少销售成本，提高效率，增强企业的核心竞争力。

零售业和客户之间的关系是一种持续不断的发展关系，一般来说零售业通常通过以下三种方法来维持和加强这种关系：尽量延长保持这种关系的时间、尽量多次地与客户交易，尽量保证每次交易的最大利润。在很多情况下商家可以比较容易地得到关于老客户的丰富的信息。这些信息特别是以前购买行为的信息中，可能包含着这个客户决定他下一个购买行为的关键信息，甚至是决定性因素。通过收集、加工和处理能够处理客户消费行为的大量信息，来确定特定消费群体或个体的兴趣消费习惯、消费倾向和消费需求，进而推断出相应消费群体或个体下一步的消费行为，然后以此为基础对所识别出来的消费群体进行特定内容的定向营销。这与传统的不区分消费对象特征的大规模营销手段相比，大大节省了营销成本，提高了营销效果，从而为企业带来更多的利润。在市场经济比较发达的国家和地区，许多公司都开始在原有信息系统的基础上通过数据挖掘对业务信息进行深度加工，以构筑自己的竞争优势，扩大自己的市场份额。

各个零售企业还可通过从销售记录中挖掘相关信息，发现购买某一种商品的顾客可能购买其他商品，这类信息可有利于形成一定的购买推荐，或者保持最佳的商品分组布局，以帮助客户选择商品，刺激顾客的购买欲望，从而达到增加销售额、节省顾客购买时间的目的。典型的成功应用案例是全球最大的连锁零售企业沃尔玛公司通过“购物篮分析”得出的“跟尿布一起购买最多的商品竟然是啤酒”的结论，从而将原来相隔很远的妇婴用品区与酒类饲料区的空间距离拉近，并适当调整价格与一定的促销手段，使得尿布与啤酒的销量双双大增。

各个零售企业往往通过办理会员卡的方式来进行客户关系管理，其目的是可以更低成本、更高效率满足客户的需求，从而可以最大限度地提高客户满意度以及忠诚度，挽回失去的客户，保

留现有的客户，不断发展新的客户，发掘并牢牢地把握住能给企业带来最大价值的客户群。

数据挖掘在寻找潜在客户方面最重要的工作是：识别好的潜在客户（定义具有什么特征的客户是好的潜在客户，找出能够瞄准具备这些特征的人群的方法）、针对不同类型的潜在客户选择合适的沟通渠道（公共关系、广告、定向市场营销）、针对不同类型的客户提供恰当的信息（同一产品，不同的人可能对不同的功能感兴趣）。

利用数据挖掘技术可以对客户群体进行划分，发现客户的不同价值和即将流失的客户以及客户流失的原因，因而就可以留住好的客户，淘汰差的客户。通过对顾客会员卡信息进行数据挖掘，可以记录一个顾客的购买序列，从而利用序列模式挖掘，可以分析顾客的消费或忠诚度的变化，据此对商品价格和花样加以调整和更新，以便留住老顾客，吸引新客户。数据挖掘技术还可以利用上述交易数据来识别顾客购买行为，发现顾客购买模式和趋势，在此基础上改进服务质量，提高货品销量比率，设计出更好的货品运输与分销策略，从而减少商业成本，取得更好的顾客保持力和满意程度。

1.7.3　在电信业中的应用

数据挖掘在电信业的应用包括：①对电信数据的多维分析；②检测非典型的使用模式，以寻找潜在的盗用者；③分析用户一系列的电信服务使用模式，来改进服务；④需求分析等。

目前，电信业有四大问题亟须解决：第一个是市场细分，即客户的分类；第二个是精确营销，即当某一个用户用了这方面的业务，他是否还会用其他的业务；第三个是新业务响应，当你推出一个套餐、新业务时，哪一类的客户会响应；第四个是客户流失，即哪一类的客户会流失，流失原因是什么，怎样预测他们的动向。

客户细分的目标可以概括为：通过对客户的人口统计特征、各业务消费特征等信息的有效挖掘和分析，制定适宜的营销策略、广告策略、促销策略、渠道策略等来实现公司的服务客户，增加企业的语音业务和各增值业务的使用量和收入的目的。最出名的客户细分是中国移动的动感地带用户的确定。通过数据挖掘分析出年龄在 25 岁以下，在校学生，有一定彩铃和上网的需求，容易接受新鲜事物的年青一代消费群体将成为未来移动通信最大的增值群体。因此，将业务为导向的市场策略率先转向了以细分客户群体为导向的品牌战略，锁定 15～25 岁年龄段的学生、年轻白领，打造新的增值市场，事实证明，锁定这一消费群来主打自己的新品牌，使中国移动动感地带品牌获得了巨大成功。

精确营销是一个基于数据分析的量化过程，对用户使用行为和偏好的精确衡量和分析，从而实现在合适的时间、合适的地点精确推荐给合适的人。

以电信运营商的彩铃为例。通过关联规则挖掘，可以得出结论：下载过周杰伦歌曲的用户中，同时下载过王力宏的比例最高，林俊杰次之。因此，可以针对下载过周杰伦歌曲的用户推荐王力宏或者林俊杰的歌曲交叉销售。

现在电信业已经迅速从单纯的提供市话和长话服务演变为综合电信服务，如语言、传真、移动电话、图形、电子邮件、互联网接入服务。电信市场的竞争也变得越来越激烈和全方位化。目前不管是住宅电话还是移动电话，每天的使用量是很大的，对于电话公司来说，如何充分使用这些数据，为自己赢得更多的利润就成了主要问题。例如移动电话中对本地和外地每分钟收多少钱对电话公司是合算，而且还能保持住自己的顾客源不被其他电话公司吸引走；怎样划分高峰时间和非高峰时间并给予不同价格最合理等，这些问题都可以通过数据挖掘来解决。

号簿管家是中国移动推出的一个专业服务于移动电话用户的通讯录业务，通过 Web、WAP、SyncML 等多种方式，为移动电话用户提供最为便捷、安全有效的个人地址服务。利用数据挖掘技术对现有的用户进行分类分析，可以得到各类用户对号簿管家新服务的购买意愿，从而灵活地对各客户分组进行宏观观察和微观细分，为现有及潜在的用户提供更加周到的服务，以稳定或获得更多的客户。

1.7.4　在管理中的应用

现代企业的竞争归根结底是人才的竞争。企业人力资源管理部门面临庞大繁杂的员工数据，要想有效地提供人力资源管理的效益，从人才配备的角度确保企事业战略目标的实现，传统的管理办法和思想越来越不能满足这个要求。有鉴于此，需要采用新的数据处理技术。数据挖掘技术可以解决以下的问题：求职应聘者的哪些关键因素最有助于企业的成功？员工某些素质的提升是否与他们业绩的提升有明显的关联？是否福利的不同选项明显影响员工队伍的稳定？是否某些特定的受教育的程度明显地最切合企业的发展？本企业中最有代表性的最合理的职业发展道路是怎样的路径？员工的提升过程与服务年限是否有明确的关系？哪些个人品质可以确保一个员工成为合格的在家上班者？缺勤与工作业绩是否有必然联系？企业是否有必要提供员工的日托服务？提前退休计划是否能为企业带来好的效益？

上述的每个问题，都可以作为采用数据挖掘技术来得到有益的回答，从而实现优化招聘、绩效考核与评估等过程的优化，以吸引并保留经验丰富的员工队伍。

数据挖掘技术在物资资源管理中也能发挥很大的作用。通过对供应链中从供应商到最终消费者的物流、信息流、资金流等各种数据的挖掘，可以有计划、有协调和有控制地管理供应链，使得供应链上的各企业成为一个协调发展的有机体，从而建立一个有竞争力的物资供应链。在这其中库存问题是首要解决的一个问题。需要根据客户的需求历史或者生产计划等求出需求规律，解决需求预测中的数据特征难以量化，需求订货周期难以确定、供应难以确定等技术问题，从而较为精确地预测客户下一时期的物资需求品种和需求量，降低供应链的成本。

1.7.5　在化学研究领域中的应用

经过两个多世纪的积累，特别是 20 世纪后合成化学的大发展，已经收集了大量功能分子信息，包括其合成、结构、性质等，现在还在以越来越快的速度合成出新化合物，堆起一座各种各样的物质信息大山。同时随着化学分析技术的进步，化学实验数据量也迅猛增长。此外，随着计算化学的发展，计算获得的数据量也相当可观。“海量”的化学数据，只有通过计算机技术以及相应的数据挖掘技术才能为科学界共享。

化学界有 CAS、Beistein 和 Gmelin 三大数据库系统。CAS 系统包括化学文献、分子结构、化学反应等各种数据；Beistein 主要是处理有关有机化学的结构与数据；Gmelin 则主要处理无机物、金属有机物结构与性质。其他类型的数据库也不断涌现，例为研究小分子在生物体系中的作用建立的 ChemBank 数据库。该数据库有 2000 多种小分子的生物活性数据；美国国家癌症研究院（NCI）将其测试过的 125000 种非专用化合物制成文本文件，每个化合物都有 CAS 登录号。MDL 将该数据库进行了转换，以便与 ISIS 集成，并能生成三维模型。SpecInfo 数据库中包含了 15 万多个化合物的 66 万多条光谱数据（核磁共振谱、近红外和质谱等数据），还含有其他实验信息和参考方面等相关信息，实现了与 CAS、Beistein、Gmelin 和 NUMRIGUIDE 等数据库的对接。

化学数据库的建立，为化学数据的高效存储、检索、集成、应用提供了方便。化学知识往往隐含在大量数据中，挖掘出这些知识需要一些思想和方法。化学信息挖掘是理论与计算机化学、环境科学、药物化学、生命科学等学科中非常重要同时又远未解决的问题。

化学数据挖掘技术从其内容看，是一个由多学科交叉形成的研究和应用领域。技术关键在于利用计算机技术、数学模型和化学背景知识等从海量化学数据中自动发现、揭示和表征那些原来不明显、具有潜在应用价值的新知识和新信息。目前，针对化学化工数据的挖掘技术均未形成与成熟，亟待研究开发。例如合成路线设计是化学家长期以来的理想，尽管已经有大量数据可供参考，但由于化学反应体系的复杂性，难以用纯理论方法来解决合成路线设计问题，只能从已知知识中找出规律，或从类比推测。基于前者建设了合成反应数据库，基于后者形成计算机辅助合成设计系统。该领域当前研究方向有反应数据挖掘和反应知识发现、反应知识模型的表述和反应知识库的建立、化合物反应性能的预测以及化学知识的类比推理等。

随着各种物理方法和物理化学方法在化合物结构分析中的推广应用，质谱、光谱、色谱、电子能谱等谱图的解析已成了比较专业的学问，不仅需要较深的理论功底，而且也需要丰富的实践经验。各种谱图包含有大量化学信息，不但可以用来鉴定未知物的成分，测定某些成分的含量，而且可以用来探讨或确定分子或固体的结构、化学键的特征等。理想的做法应当是彻底弄清各种谱图产生的机理，从而从理论上完成从实测谱图到化学成分、分子结构、化学键特征等化学信息的交换。但实际上很难完全做到这点。即以最简单的光谱——原子光谱为例，重原子的原子光谱迄今为止多数谱线不能从理论上解释。这样就不得不用经验方法对谱图做鉴别和解析工作，以达到化学分析和结构分析的目的。由于化合物种类庞杂，谱图的数据亦急剧增加，单凭少数有经验的专家来做谱图解析已不能满足需要。随着计算机技术、人工智能、数据库技术的发展，利用计算机做谱图解析的各种方法应运而生。其中有一类方法是数据库谱图显示方法，即将大量已知化合物的谱图存入数据库，通过检索的方法来识别谱图。另一类方法是利用数据挖掘技术，它利用已知谱图做训练点，对未知物的图谱作分类、鉴别以至结构测定等。由于化合物种类庞杂、数目很多且每年都在大量增加，单纯依靠已知谱图的储存和检索不能完全解决谱图解析问题。由于数据挖掘技术有某种“举一反三”的功能，能从大量已知化合物的谱图做分类工作，所以在谱图解析方面有重要的实际应用意义。迄今为止，利用数据挖掘可以对质谱、原子光谱、红外光谱、拉曼光谱、核磁共振谱、$\gamma-$射线谱、色谱等的谱图进行识别，并不同程度地收到效果，这方面的研究工作是现代分析化学的前沿课题。

1.7.6　在材料研究、生产方面的应用

金属等各种材料具有不同的性质，人们往往根据其性能确定它的用途。但是寻找一种新的材料的工作是十分艰苦的。一般要通过大量的“配方炒菜”式的实验工作，才能筛选出较好的材料。以高温合金为例，试制一种新的高温合金要初筛千百种配方，初选后还要做成千小时的高温长期性能测试。这一类先搞大批“配方炒菜”，再逐一测试性能的工作方法会消耗大量人力、物力和时间。如何利用计算机信息处理方法使寻找新材料的工作方式有所改进，以收到事半功倍的效果，是近数十年来许多科学家努力研究的课题。

瑞典钢铁公司试制了 15 种新钢种，在新钢种的钢材加工过程中，有 9 种钢材开裂，另有 6 种不开裂。为了查明钢中微量元素对钢材开裂的影响，他们分析了这 15 种钢材中的 17 种微量元素，并用数据技术中的分类算法寻找规律。结果发现：“好钢”的成分代表点集中在一个较小的

区域，可包括在一个高维空间的包络面内；而“坏钢”的数据点则很分散。这是因为引起开裂的原因不止一种，所以“坏钢”区事实上是多个区域的叠加。

数据挖掘将成为材料研究工作者不可缺少的工具，实验工作者将利用它整理实验数据，从实验数据中最大限度地提取信息，会使人变得“聪明”些，少走弯路，快些获得成功。理论工作者将利用它寻找经验规律，从中得到新的启发。工程技术人员可以用它总结生产控制、分析检验中获得的数据和经验，有利于改进生产和技术管理。

数据挖掘技术在制造业应用的需求主要是产品需求分析、产品故障诊断与预测、精确营销和工业物联网分析等。通过数据挖掘能够使客户参与到产品的需求分析和产品设计中，为产品创新做出贡献。现代化工业制造生产线安装有数以千计的小型传感器，来探测温度、压力、热能、振动和噪声。因为每隔几秒就收集一次数据，利用这些数据可以实现很多形式的分析，包括设备诊断、用电量分析、能耗分析、质量事故分析（包括违反生产规定、零部件故障）等。数据挖掘技术在材料生产过程特别是生产流程（流程工业）中也起着非常重要的作用。生产流程是指生产连续不间断或半间断批量生产的工业过程，如炼油、化工、电力、冶金、造纸等行业，其共同特点是工艺流程基本不变，但生产周期长、生产过程复杂、工艺参数特别多。随着流程工业自动化、数字化水平的不断提高，数据越来越丰富，这就为应用数据挖掘技术提供了良好的契机。应用数据挖掘技术可以将产品的生产成本、产品质量控制等生产过程优化。

产品质量和信誉是现代企业的生命线，许多产品的质量问题要在长期使用中才能显露出来。为了保证产品质量的可靠性，不仅要把握好产品检验关，更为重要的是产品生产流程中的质量控制。一般产品生产是多工艺生产，各个工艺都有影响产品质量的因素。如影响钢材表面质量的因素有：元素成分含量、铸坯的厚度及宽度、挖掘温度、铸坯拉速、时间等。假定产品质量指标有多个影响因素，数据挖掘的目的是根据对产品质量的影响因素和产品质量指标的测量数据，找出这两者之间的函数关系式或模式。然后根据得到的模式，既可以对新的工况参数，推断其对应的质量指标，即产品质量预测；也可以根据指定产品质量目标值反推相应的影响因素值，即逆质量问题；还可以据此关系模式找到降低原料、燃料消耗的方法。

1.7.7 在机械故障诊断与监测中的应用

无所不在的传感器技术的引入使得产品故障实时诊断和预测成为可能。机械设备运行状态监测和故障诊断最本质的工作是：如何通过对机器外部征兆的监测取得特征参数的正确信息，并进行分析和识别。从本质讲很明显，机械设备故障诊断与监测就是数据挖掘的应用过程，但与一般的数据挖掘应用相比，也具有几个特点：一是学习样本集中，正常运行模式样本多而故障运行模式少；两类误判会产生不同程度的损失，一般情况下将正常运行模式判为故障运行模式即错判所造成的损失远比故障模式判为正常运行模式即漏判所造成的损失小；三是生产设备投产运行一段时间内所表现出的状态一般仅有正常运行模式一种，随着时间的推移，其他运行模式才可能相继出现；随着生产设备的长时间使用和其他一些因素的出现，设备的运行参数会发生改变，因此各运行模式之间的划分标准可能改变；设备运行状态监测和故障诊断中存在较强的模糊性；诊断理论具有广泛的通用性而具体样本数据和各种参数适用面却很窄。

因此，目前常用的故障检测与诊断方法主要有：门限检测方法、信号处理方法、专家系统方法、故障诊断树方法、模式识别方法、模糊数学诊断方法、人工神经网络诊断方法和信息融合的方法等。在这些方法中，有些算法需要足够的典型故障样本，有些过于对参数摄动、噪声干扰等

因素敏感，有些由于受随机过程干扰以及各种瞬态过渡过程的存在，使得应用受到限制或及时性和准确性方面存在一定困难。利用跨国公司客户服务数据库中的服务数据可以突破数据匮乏瓶颈；改进算法或新算法可以克服算法的不足。

1.7.8 在医疗领域中的应用

在医疗领域中，大量的数据可能已存在多年，例如病人、症状、发病时间、发病频率以及当时的用药种类、剂量、住院时间等。利用这些数据挖掘可以得到许多成绩，如：心电图和心电向量图的分析；脑电图的分析；染色体的自动分类；癌细胞的分类；疾病诊断等专家系统；血相分析；医学图片的分析，包括 X 光片、CT 片等图像的分析等等。

数据挖掘在医学上的应用很多，前景广阔。通过数据挖掘不仅可以对疾病进行诊断，而且还可以进行疾病的预测。随着卫生保健事业的发展和人们生活水平的提高，健康普查将越来越普遍而且更加常态化，普查的内容也越来越丰富，单纯依靠人工分析和判断普查结果显然不能满足要求，数据挖掘技术将发挥更大的作用。

微量元素的比例失衡是许多病（尤其是地方病）的病因或重要因素。微量元素硒的防癌作用近年来受到广泛关注。同时也发现其他几种元素对硒有拮抗作用。为了查明多种微量元素对癌症发病率的影响，对 25 个国家和两个地区的居民（通过饮食）对硒、锌、镉、铜、铬、砷的平均摄入量为特征量构成多维空间，将这些国家或地区的癌症死亡率记入其中，作分类分析，可以看出乳癌高发病国家和乳癌低发病国家分布在不同区域，其间有明显的分界线。

用数据挖掘方法研究肺癌早期诊断问题，也获得显著成功。取大量的人的头发分析硒、锌、镉、铜、铬、砷、铅、锡 8 种微量元素，考察并收集其中与肺癌有关的信息并用分类方法处理，发现其中硒、锌、镉、铬、砷 5 种元素与肺癌有关。

除癌症诊断外，数据挖掘技术还可应用于其他临床化学课题。许多病都靠多种化验数据诊断得出，数据挖掘可用于化验数据自动化解释工作。例如区别甲状腺功能的三种情况，证明只用两组实验即可区别，而不是如同以前人们认为的那样需一种实验结构才能判断。

在药物实验中，可能有很多不同的组合，每种若均加以实验，则成本太大，数据挖掘技术可以大大减少实验次数以节省成本。生物医学的大量研究大都集中在 DNA 数据的分析上，人类大约有 105 个基因，一个基因通常由成百个核苷按一定序列组成，核苷按不同的次序可以组成不同的基因，几乎是不计其数，因此，数据挖掘成为 DNA 分析中的强力工具，如对 DNA 序列间的相似搜索和比较；应用关联分析对同时出现的基因序列的识别；应用路径分析发现在疾病不同阶段的致病基因等。

以上只是列举了数据挖掘的一些应用，而不是包罗万象。随着各有关交叉学科的进一步发展，数据挖掘也一定能够进一步在完善自己的理论基础上得到各行各业的进一步有效应用，越来越多的数据挖掘算法从原领域中分化出来，并正在形成一个学科。可以预见，在不久的将来，结合了数据挖掘的决策系统将为各行各业发展起到不可替代的关键作用。以下几个事件足以说明数据挖掘技术的价值和前景。

（1）有关学者撰文指出，门户解决了 Web 0.5 时代的信息匮乏；Google 解决了 Web 1.0 时代的信息泛滥；Facebook 解决了 Web 2.0 时代的社交需求；未来将是谁的十年？展望 Web 3.0 时代，当高效的社交网络趋于信息量爆炸时，我们庞大的社交关系也需要一个“Google”来处理，那就是下一个十年，数据挖掘的十年，网络智能的十年。

（2）2005 年微软将“互联网搜索、数据挖掘与语音技术”确定为亚洲研究院的三大重点研发领域。

（3）美国 2008 年评选的 12 个最有前途的职业中数据挖掘师排名第四。

（4）包括 IBM 在内的世界数据库厂商，纷纷在数据挖掘领域加大投入，把数据挖掘功能集成到其产品中，以提高产品的竞争力，2009 年 10 月 2 日 IBM 成功收购了 SPSS Inc，微软也在其 SQL Server 2005、Excel 2007 中嵌入了数据挖掘功能。

（5）国际数据公司的研究表明，数据领域存在着 1.8 亿 GB 的数据，企业数据正在以 55%的速度逐年增长。如今，只需两天就能创造出文明诞生以来至 2003 年所产生的数据的总和。“大数据”已经成为重要的时代特征。麦肯锡全球研究机构 2011 年 5 月在《大数据：创新、竞争和生产力的下一个前沿领域》中指出，充分利用大数据可帮助全球个人定位服务提供商增加 1000 亿美元的收入，帮助欧洲公共部门的管理行业每年提升 2500 亿美元产值，帮助美国医疗保健行业每年提升 3000 亿美元产值，并可帮助美国零售业获得 60%以上的利润增长率。由此可见，充分利用大数据和挖掘大数据商业价值将为行业、企业带来强大的竞争力。

数据挖掘有广泛的应用领域，但数据挖掘技术并不是万能的，同时也遇到了一些难题。银行、零售业等行业的数据挖掘必然要涉及消费者个人隐私问题，这样就会带来一些社会问题。如何避免不必要的与消费者之间的纠纷，合理利用消费者数据等，是当前数据挖掘面临的问题。

数据库也变得越来越庞大、越来越难操纵。特别是大企业、高层政府部门的数据库，这些单位往往拥有十几种甚至几十种数据，数千个数据库表，并且还可能存储在数家企业提供的分布式数据库系统中，如何从中得出有用的信息对数据挖掘提出了严峻挑战。

数据挖掘的工业标准还正处于形成过程中。具有一个好的数据挖掘工业标准将有助于数据挖掘系统平台开发的标准化，有助于方便地支持交互式的数据挖掘和灵活有效的知识发现。

数据挖掘不会替代有经验的商业分析师或管理所起的作用，毕竟它只是提供一个强大的工具，而不是有魔力的权杖。数据挖掘得到的预测模型可以告诉你会如何，但不能说明为什么会如此。数据挖掘不能在缺乏指导的情况下自动地发现模型或模式，因此在开始任何数据挖掘项目之前，必须回答一个重要的问题：是否真的需要用数据挖掘技术？要对此做出决定，重要的是理解所需的数据挖掘技术的复杂度级别，例如是否需要一个标准的打印好的报表，还是需要交互式的联机分析处理（Online Analytical Processing，OLAP）来分析数据的各种特征以及是否需要用真正的数据挖掘技术来建立预测模型、搜索数据库以获得有用的模式。选择一种数据挖掘技术和某种数据挖掘产品的关键在于产品能否带来商业价值，否则一般的数据分析就足够了。

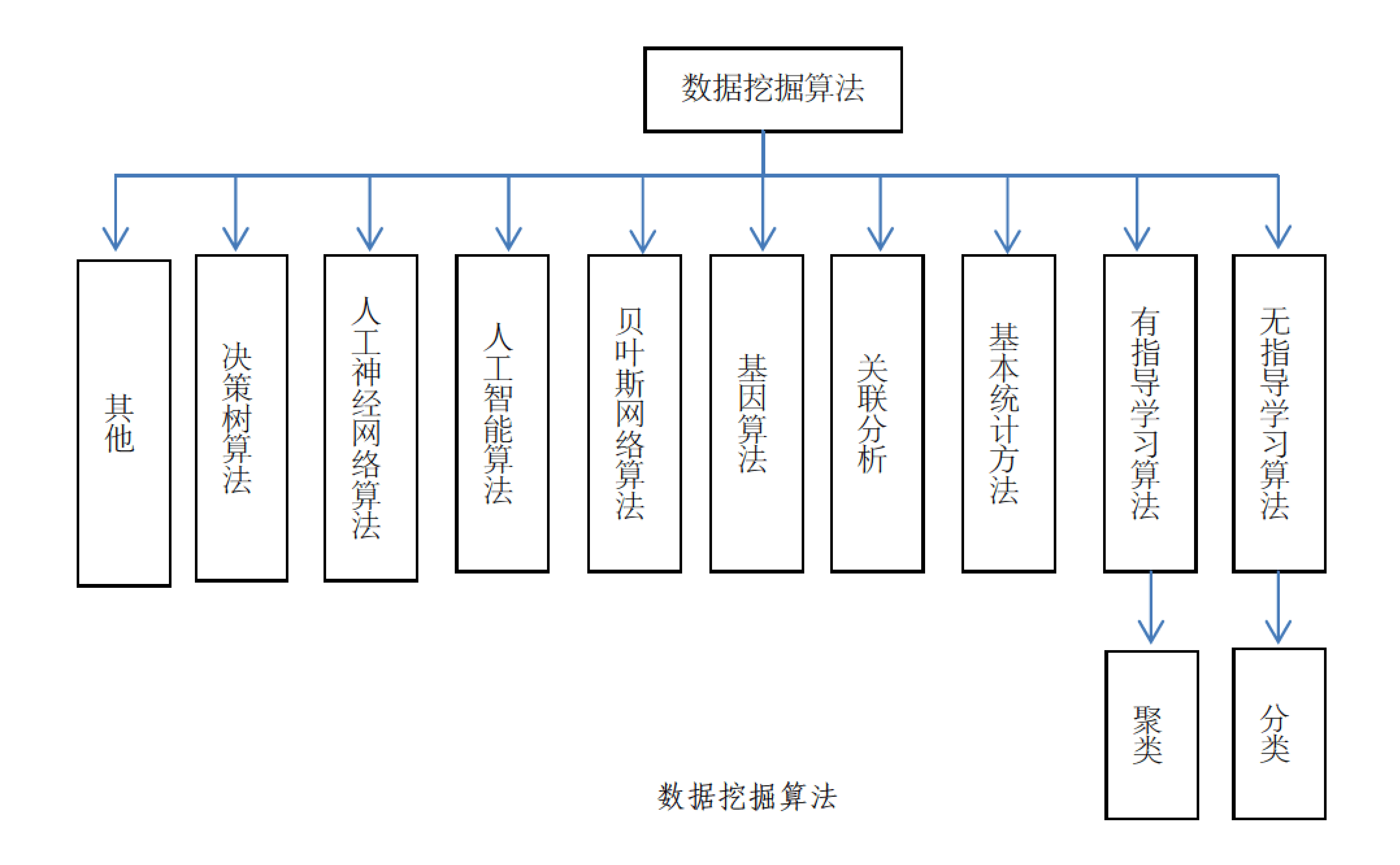
第 2 篇　数据挖掘算法

数据挖掘任务有很多实现方法，这些方法不仅需要选定的数据结构，而且需要特定的算法。一个好的算法应是兼顾效率和准确性的。一种准确性较高但耗时巨大（以天为计）的算法是不能应用于数据挖掘中的，而且算法必须同时对训练样本和测试样本都有较好的预测准确性，不能产生“过拟合”现象。

数据挖掘算法根据得到的模型特点可以分成两类，即参数模型算法和非参数模型算法。参数模型用带参数的代数方程来描述输入与输出之间的关系，其中有些参数是选定的。方程中的选定参数由输入实例确定。尽管参数模型是一个很好的理论论题并且有时也能应用于实际，但它常常过于简单，或者对涉及的数据要求过多的、无法获得的知识，因此，对于现实世界中的问题来说，这些参数模型可能是不实用的。

与参数化方法相比，非参数化方法更适合于数据挖掘应用。非参数模型是数据驱动的模型，它不使用显式的方程来确定模型。这就意味着建模过程更适用人工处理的数据。非参数化方法不像参数化方法那样事先确定一个特定的模型，而是依据输入的数据创建模型。参数化方法在建模前需要更多的有关数据的知识，而非参数化方法则需要大量的数据作为建模过程本身的输入，然后通过筛选这些数据来创建模型，近来的非参数方法已经能够应用机器学习技术在输入数据时进行动态的学习，因此数据越多，创建的模型就越好。另外，这种动态学习过程允许随着数据的输入持续地创建模型。这些特征使非参数化方法尤其适用于有大量动态数据变更的数据库。非参数方法包括人工神经网络、决策树和遗传算法。

数据挖掘包含很多算法，如下图所示。具体使用哪种算法，要根据具体情况和应用要求而定。一种数据挖掘算法可能在一种情况下适用，而在另一种情况下就不适用。在特定的应用环境下，我们应能找出最适用的数据挖掘算法，并加以实施。可以看出，数据挖掘本质上就是数学建模，即发现客观事物的规律。



数据挖掘算法

第 2 章

决策树算法

2.1 决策树算法概述

决策树是一种用于分类、聚类和预测的预测型建模方法，它采用“分而治之”的方法将问题的搜索空间分为若干子集，其形式类似于流程图。其中，每个内部节点表示在一个属性上的测试，每个分支代表一个测试输出，而每个树节点存放一个类标号。树的最顶层节点是根节点。决策树也可解释为一种特殊形式的规则集，其特征是规则的层次组织关系。决策树可以由分析训练数据的算法创建，或者由领域专家创建。大多数决策树技术随树的创建过程不同而不同。

决策树的原理与“20 问”游戏类似。图 2.1 为游戏的步骤，这棵树的根是所问的第一个问题，树中每层由游戏中这一阶段的问题组成，叶节点代表成功地猜到了希望预测的对象，它表示这是一个正确的预测。

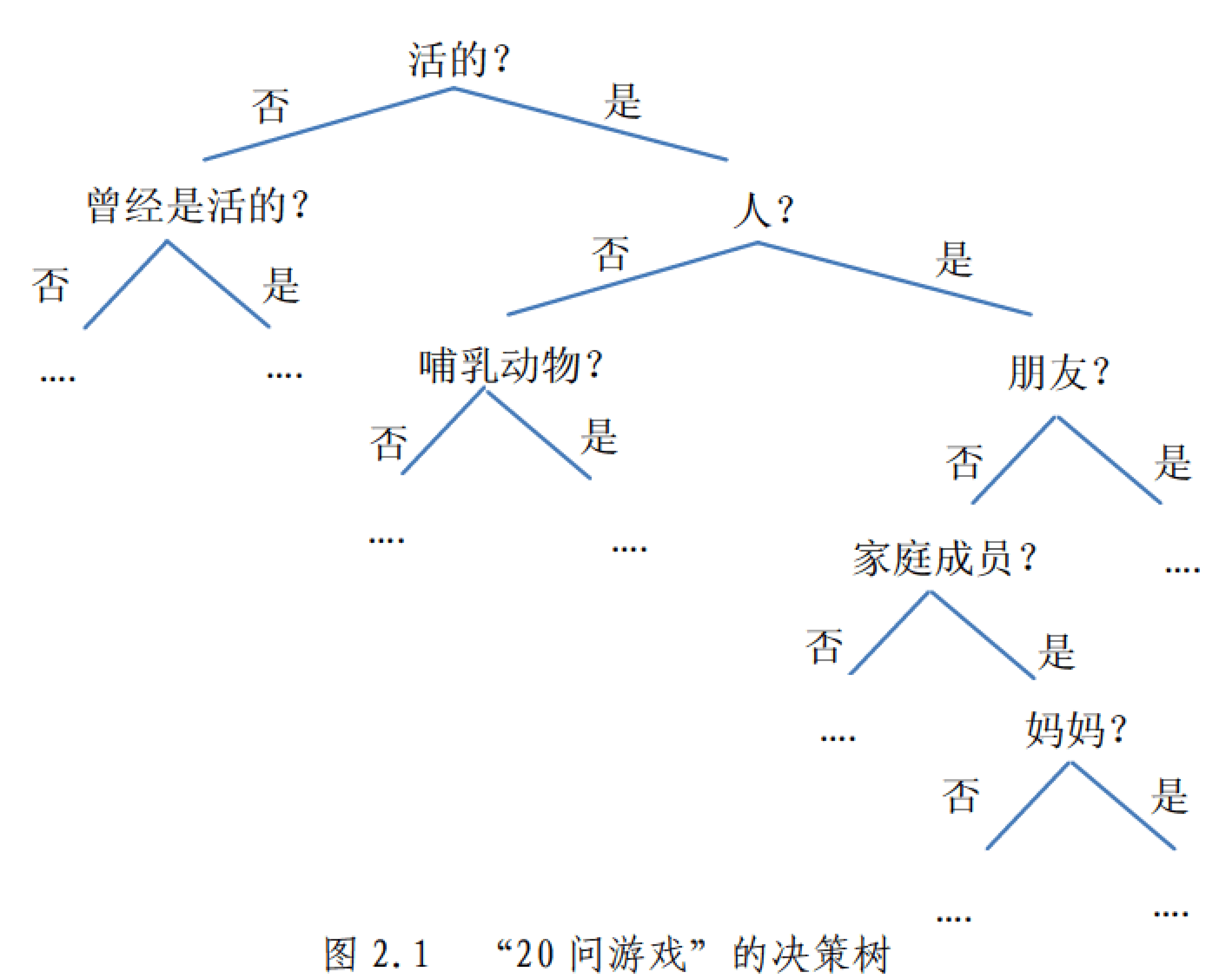


图 2.1 “20 问游戏”的决策树

决策树是主要针对“以离散型数量作为属性类型进行分类”的学习方法，对于连续型变量，必须被离散化后才能进行学习和分类。

决策树算法具有一些优点。决策树的构造不需要任何领域知识或参数设置，因此适合于探究式知识的发现。决策树可以处理高维数据。获取的知识树的形式表示是直观的，并且容易被人理解。决策树学习的归纳和分类步骤是简单和快速。一般情况下决策树具有很好的准确率。但也存在一些缺点，如决策树算法不易处理连续性数据；数据的属性域必须被划分为不同的类别才能处理，有时这样的划分比较困难；决策过程忽略了数据库属性之间的相关性等；在处理较大数据库时算法的额外开销较大，降低了分类的准确性；数据复杂性提高，分支数增加，管理的难度会越来越大。

2.2 决策树基本算法

一棵决策树的内部节点是属性或者是属性的集合，而叶节点就是学习划分的类别或结论，内部节点的属性称为测试属性或分裂属性。

当通过一组样本数据集的学习产生了一棵决策树后，就可以对一组新的未知数据进行分类。使用决策树对数据进行分类时，采用自顶向下的递归，对决策树内部节点进行属性值的判断比较

根据不同的属性值决定走向哪一条分支，在叶节点处就得到了新数据的类别或结论。图 2.2 就是一棵决策树，其中 A、B、C 表示属性名，a_1、a_2、b_1、b_2、c_1、c_2 分别表示属性 A、B、C 的取值。当属性 A 的取值为 a_1 时，属性 B 的取值为 b_2，它属于第二类。

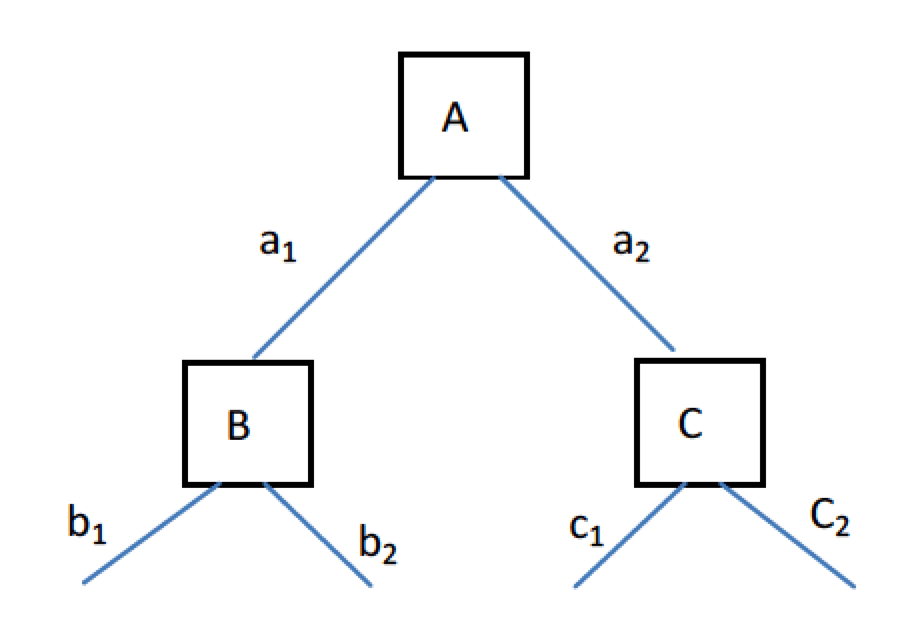


图 2.2 简单决策树

根据决策树的内部节点的各种不同的属性，可以将决策树分为以下三种。

（1）当决策树的每一个内部节点都只包含一个属性时，称为单变量决策树；当决策树存在包含多个变量的内部节点时，称为多变量决策树。

（2）根据测试属性的不同属性值的个数，可能使得每一个内部节点有两个或者是多个分支。如果每一个内部结构只有两个分支则称为二叉决策树。

（3）分类结果可能是两类也可能是多类。二叉决策树的分类结果只能有两类，故也称为布尔决策树。

从图 2.2 中可看出，决策树算法通常分为两个阶段，即树的构建阶段和树的修剪阶段。在树的构建的过程中，计算分支指标（splitting index，SI）是关键。不同的决策树算法采用不同的分支指标，ID3、C4.5 使用的分支指标是信息增益（information gain），而 CART 算法、SLIQ 算法和 SPRINT 算法使用 gini 指标。这些指标值决定了在哪个属性处发生分裂。

剪枝的目的是降低由于训练集中存在噪声而产生的起伏使得决策树产生不必要的分支，从而导致在使用决策树模型时对实测样本实施分类中出错。

大多数决策树算法都需要面临下列问题：

- 选择分裂属性：在构建决策树的过程中，哪个属性作为分裂属性会影响算法性能。属性的选择不仅涉及检验训练集中的数据，而且还需要参考领域专家的建议。
- 分裂属性的次序：选择分裂属性的次序也是很重要的。较好的分裂次序可以减少算法量。
- 分裂：与分类属性的次序相应的是确定分裂的数目。分裂的数目要根据属性的定义域来确定。
- 树的结构：为了改进应用树进行分类的性能，总是希望得到具有最少层次的平衡树。
- 当训练数据被正确分类时，树的产生过程就应停止。为了防止产生过大的树或产生过拟合，有时也希望提前停止。提前停止指标需综合考虑分类精度和性能等多个因素。
- 训练数据：产生的决策树的结构取决于训练数据。如果训练数据集太小，则产生的树由于没有足够的特殊性，而不能很好地应用于更加通用的数据。如果训练数据集太大，则产生的树可能产生过拟合。

- 剪枝：一棵树被构建后，还需要对树进行修剪以提高分类阶段树的性能。剪枝阶段可能会删除过多的比较或者删去一些子树，以获得更好的性能。

在设计构建决策树算法时，总是希望得到可以对数据集进行正确分类的最佳形状的树。树归纳算法和训练数据共同决定树的形状。

图 2.3 为同一问题的不同决策树。对于大型数据库通常期望得到短的平衡树。

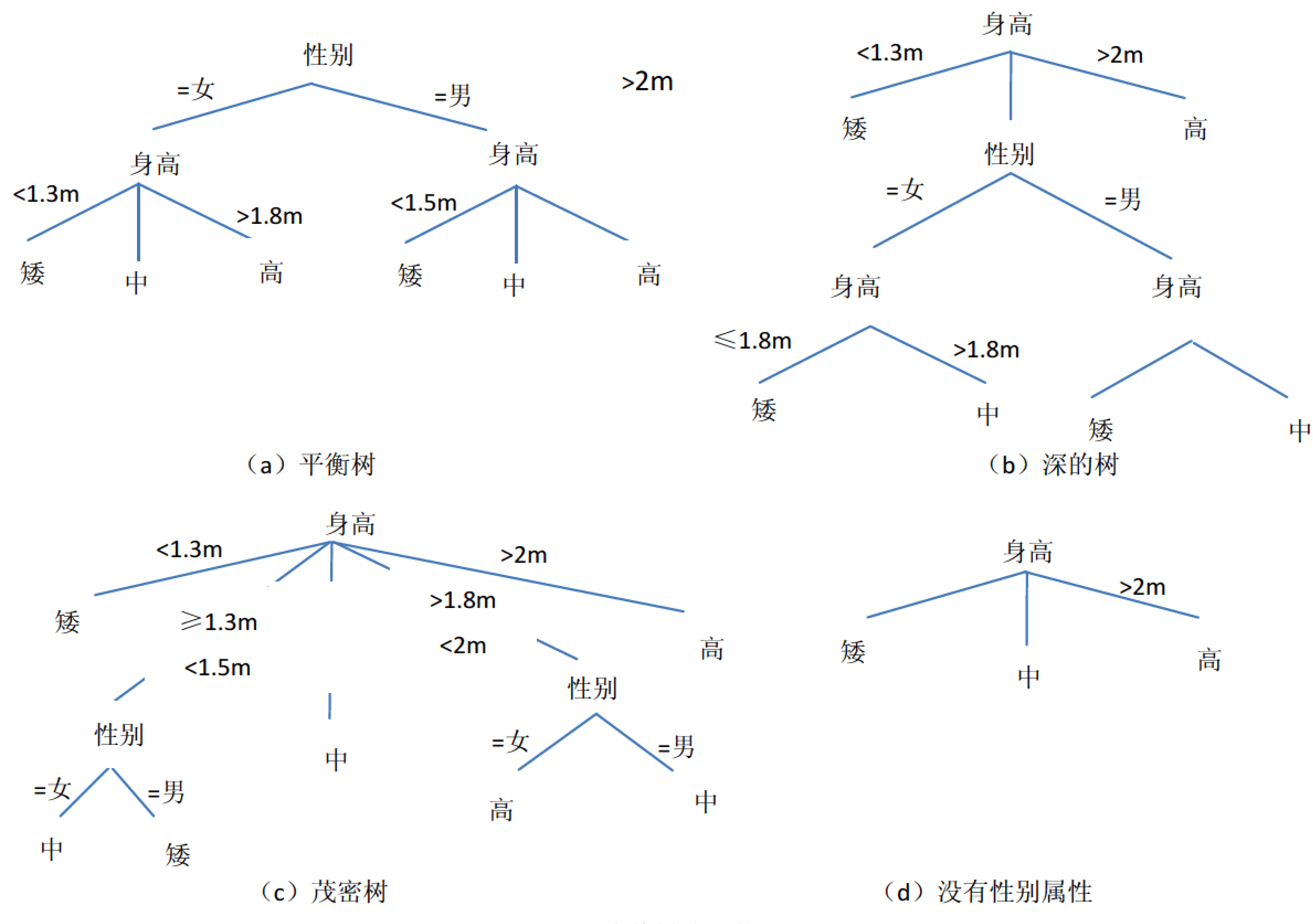


图 2.3 决策树的比较

决策树算法的时间和空间复杂性取决于训练数据的规模、属性数目以及最终产生的树的形状。在最坏的情况下，构建的决策树深度可能很深而不茂密。

2.3 ID3 算法

ID3 算法是各种决策树算法中最有影响力、使用最广泛的一种，其基本策略是首先选择具有最高信息增益的属性作为分裂属性。

设样本数据集为 X，类别数为 n。设属于第 i 类的样本数据个数是 C_i，X 中总的样本数为 $|X|$，则一个样本属于第 i 类的概率 $P(C_i)\approx\dfrac{C_i}{|X|}$。此时决策树对划分 C 的不确定程度（即信息熵）为

$$H(X,C)=H(X)=-\sum_{i=1}^{n}p(C_i)\log_2 p(C_i)$$

若选择属性 a（设属性 a 有 m 个不同的取值）进行测试，其不确定程度（即条件熵）为

$$
\begin{aligned}
H(X \mid a) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} p(C_i, a=a_j)\log_2 p(C_i \mid a=a_j) \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} p(a=a_j)p(C_i \mid a=a_j)\log_2 p(C_i \mid a=a_j) \\
&= -\sum_{j=1}^{m} p(a=a_j)\sum_{i=1}^{n} p(C_i \mid a=a_j)\log_2 p(C_i \mid a=a_j)
\end{aligned}
$$

则属性 a 对于分类提供的信息量为

$$I(X,a) = H(X) - H(X \mid a)$$

式中：$I(X,a)$表示属性作为分类属性之后信息熵的下降的程度，亦即不确定性下降的程度（亦称为信息增益），所以应该选择使得 $I(X,a)$最大的属性作为分类属性，这样得到的决策树的确定性最大。

ID3 算法的步骤如下。

（1）选出整个样本数据集 X 的规律为 W 的随机子集 X_1（W 称为窗口规模，子集称为窗口）。

（2）以 $I(X,a) = H(X) - H(X \mid a)$ 的值最大，即 $H(X \mid a)$的值最小为标准，选取每次的测试属性，形成当前窗口的决策树。

（3）顺序扫描所有样本数据，找出当前的决策树的例外，如果没有例外，则结束算法。

（4）组合当前窗口的一些样本数据与某些在（3）中找到的例外形成新的窗口，转（2）。

基本的 ID3 算法采用信息增益作为单一的属性的度量，试图减少树的平均深度，而忽略了对叶子数目的研究，导致了许多问题：信息增益的计算依赖于属性取值的数目较多的特征，而属性取值较多的属性不一定是最优属性；抗噪性差，训练集中正例和反例较难控制。因此，针对 ID 算法的不足，提出了许多改进策略。

① 离散化：在处理连续性属性时，可以将其离散化。最简单的方法是将属性值分成两段。对任何一个属性，其所有的取值在一个数据集中是有限的。假设该属性取值为 $\{a_1,a_2,\cdots,a_n\}$，首先将其值按递增顺序排列，然后每对相邻值的中点看作可能的分裂点，一共存在 $n-1$ 个分段值（即为均值，如$\dfrac{a_i+a_{i+1}}{2}$）。ID3 算法采用计算信息量的方法计算最佳的分段值，然后进一步构建决策树。

② 空缺值处理：训练集中的数据可能会出现某一训练样本中某一属性值空缺的情况，因此必须进行空缺值处理。可以用属性值的最常见值、平均值、样本平均值等代替空缺值。

③ 属性选择度量：在决策树的构建过程中，有许多的属性选择度量。可以采用其他属性选择度量来提高算法的性能。

④ 可伸缩性：ID3 算法对于相对较小的训练数据是有效的，但对于现实世界中数以百万计的训练数据集，需要频繁地将训练数据在主存和高速缓存中换进换出，从而导致算法的性能低下。因此可以将训练数据分成几个子集，使得每个子集能够放入内存，然后由每个子集构造一棵决策树，最后，将每个子集得到的分类规则组合起来，得出输出的分类规则。

⑤ 碎片、重复和复制处理：碎片是指在一个给定的分支中的样本数太少从而失去统计意义。

解决的方法是将分类属性分组，决策树节点可以测试一个属性值是否属于给定的集合；另一种解决方法是创建二叉决策树，在树的节点上进行属性的布尔测试，从而可以减少碎片。

当一个属性沿树的一个给定的分支重复测试时，将出现重复。复制是指复制树中已经存在的子树，以上问题可以由给定的属性构造新的属性（即属性构造）来解决。

2.4　C4.5 算法

C4.5 算法是 ID3 算法的改进，它是在 ID3 基础上增加了对连续属性、属性值空缺情况的处理，对树剪枝也有了较为成熟的方法。

与 ID3 算法不同，C4.5 算法选取具有最高信息增益率的属性作为测试属性。对样本集 T，假设变量 a 有 n 个属性，属性取值 $a_1,a_2,\cdots,a_k$，对应 a 取值 a_i 出现的样本数分别为 n_i，若 n 是样本的总数，则应有 $n_1+n_2+\cdots+n_k=n$。C4.5 算法利用属性的熵值来定义为了获取样本关于属性的信息所需要付出的代价，即

$$H(X,a)=-\sum_{i=1}^{n}p(a_i)\log_2 p(a_i)\approx-\sum_{i=1}^{k}\frac{n_i}{n}\log_2\frac{n_i}{n}$$

信息增益率定义为平均互信息与获取信息所付出代价的比值，即

$$E(X,a)=\frac{I(X,a)}{H(X,a)}$$

即信息增益率是单位代价所取得的信息量，是一种相对的信息量不确定性度量。以信息增益率作为测试属性的选择标准，是选择 $E(X,a)$最大的属性 a 作为测试属性。

算法 C4.5 在如下几个方面优于 ID3 算法。

（1）一些样本的某些属性取值可能为空，在构建决策树时，可以将这些缺失值用最常用的值代替，或者用该属性的所有取值的平均值代替，从而处理缺少属性值的训练样本。另一种解决方法是采用概率的方法，对属性的每一个取值赋予一个概率，在划分样本集时，将未知属性值的样本按照属性值的概率分配到子节点中去，这些概率的获取依赖于已知的属性值的分布。

（2）C4.5 算法不仅可以处理离散属性，也可以处理连续属性。基本思想是按数值属性值的大小对样本排序，从中选择一个分割点，划分数值属性的取值区间，从而将 ID3 的处理扩充到数值属性上来。

（3）增加了剪枝算法。在 C4.5 算法中，有两种基本的剪枝策略：

- 子树替代法剪枝是指用叶节点替代子树。仅当替代后的误差率与原始树的误差率接近时才替代。子树替代是从树枝向树根方向进行的。
- 子树上升法是指用一棵子树中最常用的子树来代替这棵子树。子树从当前位置上升到树中较高的节点处。对于这种替代也需要确定误差率的增加量。

（4）分裂时 ID3 算法偏袒具有较多值的属性，因而可能导致过拟合，而信息增益率函数可以弥补这个缺陷。

（5）使用 k 次迭代交叉验证，评估模型的优劣程度。交叉验证是一种模型评估方法，它将使用学习样本产生的决策树模型应用于独立的测度样本，从而对学习的结果进行验证。如果对学

习样本进行分析产生的大多数或者全部分支都是基于随机噪声的，那么使用测试样本进行分类的结果将非常差。如果将上述的学习—验证过程重复 k 次，就称为 k 次迭代交叉验证。首先将所有的训练样本平均分成份，每次使用其中的一份作为测试样本，使用其余的份作为学习样本，然后选择平均分类精度最高的树作为最后结果。通常分类精度最高的树并不是节点最多的树。

但是 C4.5 算法同样存在缺点，它偏向于选择对统一属性值比较集中的属性（即熵值最小的属性），而并不一定是对分类贡献最大、最重要的属性。

2.5　CART 算法

在 ID3 与 C4.5 算法中，当确定作为某层树节点的变量属性值较多时，按每一属性值引出一个分支进行递归算法，就会出现引出较多的分支，对应算法次数也多，使决策树建树算法速度缓慢。解决这个问题的方法是建立二叉决策树，即使每个树节点只产生两个分支（二叉）。CART 算法即为这样一种算法。CART 算法确定树节点（即测试属性）与 ID3 算法一样，以平均互信息作为分裂属性的度量，对于取定的测度属性变量 a，若有 n 个属性值 $s_1,s_2,\cdots,s_n$，应选取“最佳”分裂值属性值 s_i 作为分裂点引出两个分支，以使分类结果是尽可能合理正确。“最佳”分裂属性值应满足条件

$$\Phi(s_0/a)=\max_i \Phi(s_i/a)$$

其中

$$\Phi(s/a)=2P_L P_R\sum_{j=1}^{m}|P(C_i|a_{\mathrm{L}})-P(C_j|a_{\mathrm{R}})|$$

$\Phi(s/a)$ 主要度量在节点 a 的 s 属性值引出两个分支时，两分支出现的可能性以及两分支每个分类结果出现的可能性差异大小。当 $\Phi(s/a)$ 较大时，表示两分支分类结果出现的可能性差异大，即分类不均匀，特别地，当一分支完全含有同一类别结果的样本而另一分支不含有时，差异最大，这种情况越早出现表示利用越小节点，可以越快获得分类结果。$\Phi(s/a)$ 中的 L 和 R 是指树中当前节点的左子树和右子树。P_{L} 和 P_{R} 分别代表在训练集（样本集）中的样本在树的左边和右边的概率，其计算公式为

$$P_{\mathrm{L}}=\frac{\text{左子树中的样本数}}{\text{样本总数}}$$

$$P_{\mathrm{R}}=\frac{\text{右子树中的样本数}}{\text{样本总数}}$$

P（$C_i|a$）与 P（$C_i|a$）分别指在左子树和右子树中的样本属于 C_i 的概率，其计算公式为

$$P(C_i|a_{\mathrm{L}})=\frac{\text{左子树属于　类的样本数}}{a_{\mathrm{L}}\text{节点样本数}}$$

$$P(C_i|a_{\mathrm{R}})=\frac{\text{右子树属于}C_i\text{类的样本数}}{a_{\mathrm{R}}\text{节点样本数}}$$

CART 的一大优点是它将模型的验证和最优通用树的发现嵌在了算法中。它首先生成一棵非常复杂的树，再根据交叉验证和测试集验证的结果对树进行剪枝，从而得到最优通用树。

2.6 决策树的评价标准

对于一个决策树算法，可以用以下一些性能或指标进行评价。

1. 正确性

评价一棵决策树最首要、最基本的标准就是分类正确性。只有保证较高的分类正确性，才能评价决策树的其他性能。

2. 过学习

在决策树的学习过程中，可能会得到若干棵和训练实例集相匹配的决策树，必须在它们当中选择应用于实测样本。而如果有过多的决策树和训练实例集相匹配，那么模型的泛化能力（预测准确度）将很差，这种情况称为“过学习”。

3. 有效性

估计一棵决策树在测试实例集合上的性能是通过比较它在测试实例集合上实际测试结果来完成。但是这种方法等价于在测试实例集合上训练决策树，这在大多数情况下都是不现实的。所以一般不采用这种方法，而是采取用训练实例集本身来估计训练算法的有效性。一种最简便的方法是用训练实例集合的一部分（例如 2/3 的训练实例）对决策树进行训练，而用另外一部分（另外的 1/3 的训练实例）检测决策树的有效性。但是这样将减少训练实例集合的数目而增大过学习的可能性，特别是当训练实例的数目较少时更会如此。所以一般是利用下面的交叉有效性和余一有效性来评价一个决策树学习系统的有效性。

4. 交叉有效性

在这一方法中，将训练实例集 T 分为互不相交并且大小相等的 k 个子集 $T_1,T_2,\cdots,T_k$，对于任意子集 T_i，用 $T-T_i$ 训练决策树，之后用 T_i 对生成的决策树进行测试，得到错误率 e_i，然后估计整个算法的错误率 $e=\frac{1}{k}\sum_{i=1}^{k}e_i$，可以看出随着 k 的增加，所生成的树的数目随之增加，因此算法的复杂度也会变大。

5. 余一有效性

这种有效性的度量与交叉有效性类似，不同之处在于将每一个 T_i 的大小定为 1。假设$|T|=n$，则估计整个算法的错误率 $e=\frac{1}{n}\sum_{i=1}^{n}e_i$。很明显这种有效性算法的复杂度很高，但是它的准确度也很高。

6. 复杂度

决策树的复杂程度也是度量决策树学习效果的一个很重要的标准，一般有以下三种评价标准：

（1）最优覆盖问题（MCV），即生成最少数目叶节点的决策树。

（2）最简公式问题（MCOMP），即生成每个叶节点深度最小的决策树。

（3）最优学习问题（OPL），即生成的叶子最少并且每个叶子深度最小的决策树。其中，叶节点深度是指叶节点相距根节点的层数。

2.7　决策树的剪枝及优化

在决策树创建时，由于数据中的噪声和离群点，许多分支反映的是训练数据中的异常。对于这种代表异常的分支可以通过剪枝的方法去除。

一般来说，如果建立的决策树构造过于复杂，则决策树是难以理解的，对应决策树的知识规则出现冗余，将导致难以应用。另外，当决策树越小，则存储这棵树所要花费的代价也越小，因此建立有效的决策树，不仅需要考虑分类的准确性，而且需要考虑决策树的复杂程度，即建立的决策树，在保证具有一定的分类正确率的条件下，越简化越好。

最常用的决策树简化方法就是剪枝。剪枝的原则包括：①奥卡姆剃刀原则，即“如无必要，勿增实体”，即在与观察相容的情况下，应当选择最简单的一个；②决策树越小就越容易理解，其存储与传输的代价就越小；③决策树越复杂，节点越多，每个节点包含的训练样本数越少，则支持每个节点的假设的样本个数就越少，可能导致决策树在测试集上的分类错误率越大。但决策树过小也会导致错误率越大，因此需要在树的大小与正确率之间寻找均衡点。

剪枝技术主要包括预剪枝和后剪枝。

1. 预剪枝

预剪枝就是预先指定某一相关阈值，决策树模型有关参数在达到该阈值后停止树的生长。预剪枝方法不必生成整棵决策树，且算法相对简单，效率很高，适合解决大规模问题，但预先指定阈值不易确定。较高的阈值可能导致过分简化的树，而较低的阈值可能使得树的简化太少。一般地，多以样本集应达到的分类正确率作为阈值进行预剪枝控制，此时树形的复杂度可能通过随阈值变化而确定。更普遍的方法是采用统计意义下的 χ^2 检验、信息增益等度量，评估每次节点分裂对系统性能的增益。如果节点分裂的增益小于预先给定的阈值，则不对该节点进行扩展。如果在最好的情况下的扩展增益都小于阈值，即使有些节点的样本不属于同一类，算法也可以终止。

2. 后剪枝

后剪枝就是对已生成（建立）的决策树以一定的标准进行剪枝，使决策树能简化并具有一定的分类正确率。

决策树后剪枝算法，就是针对未经剪枝的决策树，应用算法将决策树的某一个或几个子树删除，得到简化的决策树，对多种不同剪枝结果所得到的简化决策树进行评价，找出最好的剪枝形式以确定最终的决策树。其中，剪枝过程删除的子树可用叶节点代替，这个叶节点所属的类用这棵子树中大多数训练实例所属的类来代替。

后剪枝算法有自上而下和自下而上两种剪枝策略。自下而上的算法首先从底层的内节点开始剪枝，剪去满足一定条件的节点，在生成的新决策树上递归调用这个算法，直到没有可以剪枝的节点为止。自上而下的算法是从根节点开始向下逐个考虑节点的剪枝问题，只要节点满足剪枝的

条件就进行剪枝。

一般的后剪枝的方法步骤如下。

设 T_0 为原始树，T_{i+1} 是由 T_i 中一个或多个的子树被叶节点所代替得到的剪枝树。

① 第 i 次剪枝评价：若第 i 次的原始树是 $T_i,T_{i1},T_{i2},\cdots,T_{ik}$ 分别是对 T_i 的各种可能剪枝结果，可用以下评价标准选出一种最好的剪枝形式，即

$$a=\frac{M}{N(L(S)-1)}$$

式中：M 是剪枝树分类错误率增加数；N 是样本总数；$L(S)$是剪枝树被去掉的叶节点数。

② 对各次得到的剪枝 $T_1,T_2,\cdots,T_k$，用相同的样本测试其分类的错误率，错误率最小的为最优的剪枝决策树。

作为选择，对于组合方法，预剪枝和后剪枝可以交叉使用。后剪枝所需的计算要比预剪枝的多，但通常产生更可靠的树，但没有一种剪枝方法优于其他所有方法。

2.8 基于 MATLAB 的决策树分析

例 2.1 表 2.1 是有关天气的数据样本集合。每个样本有 4 个属性变量：Outlook、Temperature、Humidity 和 Windy。样本集分为两类，即 P 和 N，分别表示正例和反例。利用 ID3 算法求解其决策规则。

表 2.1 天气样本数据

属 性	Outlook	Temperature	Humidity	Windy	类 别
1	Overcast	Hot	High	Not	N
2	Overcast	Hot	High	Very	N
3	Overcast	Hot	High	Medium	N
4	Sunny	Hot	High	Not	P
5	Sunny	Hot	High	Medium	P
6	Rain	Mild	High	Not	N
7	Rain	Mild	High	Medium	N
8	Rain	Hot	Normal	Not	P
9	Rain	Cool	Normal	Medium	N
10	Rain	Hot	Normal	Very	N
11	Sunny	Cool	Normal	Very	P
12	Sunny	Cool	Normal	Medium	P
13	Overcast	Mild	High	Not	N
14	Overcast	Mild	High	Medium	N
15	Overcast	Cool	Normal	Not	P
16	Overcast	Cool	Normal	Medium	P
17	Rain	Mild	Normal	Not	N
18	Rain	Mild	Normal	Medium	N

续表

属　性	Outlook	Temperature	Humidity	Windy	类　别
19	Overcast	Mild	Normal	Medium	P
20	Overcast	Mild	Normal	Very	P
21	Sunny	Mild	High	Very	P
22	Sunny	Mild	High	Medium	P
23	Sunny	Hot	Normal	Not	P
24	Rain	Mild	High	Very	N

解：

根据 ID3 算法原理，可进行编程计算，求出其决策规则。

```
>>sample={'outlook' 'temperature' 'humidity' 'windy' 'Nan'
'overcast' 'hot' 'high' 'not' 'N';'overcast' 'hot' 'high' 'very' 'N';'overcast' 'hot'
'high' 'medium' 'N';'sunny' 'hot' 'high' 'not' 'P';'sunny' 'hot' 'high' 'medium'
'P';'rain' 'mild' 'high' 'not' 'N';'rain' 'mild' 'high' 'medium' 'N';'rain' 'hot'
'normal' 'not' 'P';'rain' 'cool' 'normal' 'medium' 'N';'rain' 'hot' 'normal' 'very'
'N';'sunny' 'cool' 'normal' 'very' 'P';'sunny' 'cool' 'normal' 'medium' 'P';'overcast'
'mild' 'high' 'not' 'N';'overcast' 'mild' 'high' 'medium' 'N';'overcast' 'cool'
'normal' 'not' 'P';'overcast' 'cool' 'normal' 'medium' 'P';'rain' 'mild' 'normal' 'not'
'N';'rain' 'mild' 'normal' 'medium' 'N';'overcast' 'mild' 'normal' 'medium'
'P';'overcast' 'mild' 'normal' 'very' 'P';'sunny' 'mild' 'high' 'very' 'P';'sunny'
'mild' 'high' 'medium' 'P';'sunny' 'hot' 'normal' 'not' 'P';'rain' 'mild' 'high' 'very'
'N'};
rule=mytree_decisionID3(sample);  %ID算法函数，限于篇幅，在此不再列出
```

求得决策规则如下。

```
>>rule={1×3 cell} {1×5 cell}{1×5 cell} {1×5 cell} {1×5 cell} {1×7 cell} {1×7 cell};
>>rule{1}= 'outlook'    'sunny'                 'P'
>>rule{2}= 'outlook'    'overcast'              'humidity'    'high'      'N'
>>rule{3}='outlook'     'overcast'              'humidity'    'normal'    'P'
>>rule{4}='outlook'     'rain'                  'temperature  'cool'      'N'
>>rule{5}='outlook'     'rain'                  'temperature' 'mild'      'N'
>> rule{6}='outlook'    'rain''temperature'     'hot''windy'  'not'       'P'
>> rule{7}='outlook'    'rain''temperature'     'hot''windy'  'very'      'N'
```

例 2.2　现在高校规模不断扩大，学生数量越来越多，随着社会的发展，影响学生学习成绩的因素越来越多，特别是高职院校的学生，他们的学习基础比较差，自制力也比较弱，影响学生学习成绩的因素也较多，学生管理工作的难度就大，因此对学生成绩的影响因素分析尤为重要。对学生日常行为进行分析，从大量数据存在的关系、规则中研究学生行为，从这些行为预测学生学习成绩的发展趋势，从而使教师对学生管理工作有的放矢，有针对性地管理好学生日常行为，

从而提高学生的学习效果就显得非常重要。

表 2.2 为训练数据集，表中的决策属性都为连续属值，所以应对其进行离散化处理，使其适合使用决策树方法。离散化处理结果为旷课时数：A 表示≤5，B 表示≤20，C 表示＞20。消费金额：A 表示≤800，B 表示≤1500，C 表示＞1500。总评成绩：A 表示良好，B 表示一般，C 表示较差。

表 2.2　训练数据集

性　别	旷课时数	是否贷款	消费金额	总评成绩
女	A	否	B	B
男	A	否	B	A
男	C	否	C	C
女	B	是	A	B
女	B	否	A	A
男	C	否	C	C
女	B	否	A	A
女	A	否	A	A
女	B	否	A	A
男	C	是	A	B
男	A	否	A	A
男	A	否	A	A
男	C	否	C	C
女	C	否	B	C
…	…	…	…	…

解：

```
>>x={'性别' '旷课时数' '是否贷款' '消费金额' 'Nan'
 '女''A' '否' 'B' 'B'; '男' 'A' '否' 'B' 'A'; '男' 'C' '否' 'C'  'C';
 '女''B' '是' 'A' 'B'; '女' 'B' '否' 'A' 'A'; '男' 'C' '否' 'C'  'C';
 '女''B' '否' 'A' 'A'; '女' 'A' '否' 'A' 'A'; '女' 'B' '否' 'A'  'A';
 '男''C' '是' 'A' 'B'; '男' 'A' '否' 'A' 'A'; '男' 'A' '否' 'A'  'A';
 '男''C' '否' 'C' 'C'; '女' 'C' '否' 'B' 'C'};
 >>rule=mytree_decisionID3(x);
```

求得决策规则为：

```
>> rule{1} = '旷课时数'    'A'    '性别'     '男'    'A'
>> rule{2} = '旷课时数'    'B'    '是否贷款'  '否'    'A'
>> rule{3} = '旷课时数'    'B'    '是否贷款'  '是'    'B'
>> rule{4} = '旷课时数'    'C'    '是否贷款'  '否'    'C'
>> rule{5} = '旷课时数'    'C'    '是否贷款'  '是'    'B'
>> rule{6} = '旷课时数'    'A'    '性别'     '女'    '消费金额'  'A' 'A'
```

```
>> rule{7} = '旷课时数'        'A'        '性别'        '女'        '消费金额'    'B'  'B'
```

此例由于训练集的样本数较少，所以决策规则有可能有所欠缺。

例 2.3　以 $\log(1/EC_{50})$作为活性高低的界限，测定了 26 个含硫芳香族化合物对发光菌的毒性数据。分别计算了这些化合物的 $\lg K_{ow}$、Hammett 电荷效应常数 σ 并测定了水解速度常数 k，试根据活性类别（两类）及变量 $\lg K_{ow}$、σ 和 $\lg k$ 所取的数据，具体数据见表 2.3，对三个未知活性同系物的活性进行判别。

表 2.3　26 个化合物的结构参数与判别分析结果

化合物编号与类别		$\lg(1/EC_{50})$	σ	$\lg K_{ow}$	pK
1	第 I 类（低活性）	0.93	1.28	2.30	1.76
2		1.02	0.81	3.61	2.43
3		1.03	0.81	3.81	2.31
4		1.12	1.51	3.01	1.98
5		1.13	1.04	4.32	2.20
6		1.18	1.28	0.98	1.30
7		1.32	1.28	2.30	2.05
8		1.37	1.23	0.98	1.09
9		1.41	1.04	4.32	2.12
10		1.43	1.51	1.89	1.17
11		1.45	0.81	2.29	1.48
12		1.51	1.04	3.00	1.40
13		1.51	1.48	0.95	0.57
14	第 II 类（高活性）	1.66	1.48	2.27	1.25
15		1.67	1.71	0.66	0.59
16		1.71	1.48	0.95	0.49
17		1.72	1.48	2.27	1.22
18		1.70	1.04	3.00	1.29
19		1.87	1.71	3.00	1.10
20		1.93	1.51	3.01	1.73
21		2.19	2.06	2.04	1.76
22		2.20	1.51	1.69	1.02
23		2.21	1.59	2.03	1.23
24		2.22	2.26	2.01	0.61
25		2.56	1.71	0.66	0.57
26		2.65	2.06	0.58	1.17
27	未知	1.33	0.81	2.29	1.71
28		1.72	1.59	3.35	1.46
29		1.55	1.71	3.00	1.17

解：

在MATLAB中，自带有决策树算函数，本例用此法进行计算，此时应注意样本数据为数值矩阵。有关此函数的用法，请查阅该函数的使用说明。

```
>>x=[0.93 1.02 1.03 1.12 1.13 1.18 1.32 1.37 1.41 1.43 1.45 1.51 1.51 1.66 1.67
1.71 1.72 1.70 1.87 1.93 2.19 2.20 2.21 2.22 2.56 2.65;
  1.28 0.81 0.81 1.51 1.04 1.28 1.28 1.23 1.04 1.51 0.81 1.04 1.48 1.48 1.71 1.48
1.48 1.04 1.71 1.51 2.06 1.51 1.59 2.26 1.71 2.06;
  2.30 3.61 3.81 3.01 4.32 0.98 2.30 0.98 4.32 1.89 2.29 3.00 0.95 2.27 0.66 0.95
2.27 3.00 3.00 3.01 2.04 1.69 2.03 2.01 0.66 0.58;
  1.76 2.43 2.31 1.98 2.20 1.30 2.05 1.09 2.12 1.17 1.48 1.40 0.57 1.25 0.59 0.49
1.22 1.29 1.10 1.73 1.76 1.02 1.23 0.61 0.57 1.17]';
>>group=[1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2]';
>>t=treefit(training,group);
>>sample=[1.33 0.81 2.29 1.71;1.72 1.59 3.35 1.46;1.55 1.71 3.00 1.17];
>>result=treeval(t,sample);
求得结果为：result=1  2  1
```

例2.4　用C4.5算法对例2.1中的实例进行分类分析。

解：

根据C4.5算法的原理，对其进行编程计算，求出决策规则。

```
>> rule=mytree_decisionC4_5(sample);  %C4.5算法计算函数，限于篇幅在此不再列出
```

得到以下的决策规则：

```
>>rule={1×3 cell} {1×5 cell}{1×5 cell} {1×5 cell} {1×5 cell} {1×7 cell} {1×7 cell};
>> rule{1}= 'outlook'   'sunny'     'P'
>> rule{2}= 'outlook'   'overcast'  'humidity'     'high'  'N'
>> rule{3}= 'outlook'   'overcast'  'humidity'     'normal''P'
>> rule{4}= 'outlook'   'rain'      'temperature'  'cool'  'N'
>> rule{5}= 'outlook'   'rain'      'temperature'  'mild'  'N'
>> rule{6}= 'outlook'   'rain'      'temperature'  'hot'   'windy' 'not'   'P'
>> rule{7}= 'outlook'    'rain'     'temperature'  'hot'   'windy' 'very'  'N'
```

例2.5　利用CART算法对表2.4中的数据进行分类分析。

表 2.4　身高样本数据

编　号	性　别	身高（m）	类　别
1	女	1.6	矮
2	男	1.6	高
3	女	1.6	中

续表

编　号	性　别	身高（m）	类　别
4	女	1.6	中
5	女	1.6	矮
6	男	1.6	中
7	女	1.6	矮
8	男	1.6	矮
9	男	1.6	高
10	男	1.6	高
11	女	1.6	中
12	男	1.6	中
13	女	1.6	中
14	女	1.6	中
15	女	1.75	中

解：

首先对数据属性进行处理。设其被划分6个子区间：（0,1.6），[1.6,1.7)，[1.7,1.8)，[1.8,1.9)，[1.9,2.0)，[2.0,∞)。利用这些区间可得到潜在的分裂值1.6、1.7、1.8、1.9、2.0。然后根据CART算法的原理，从这6个可能的属性值中选择一个分裂点。以下为CART算法的程序，限于篇幅，不再详细列出。

```
>> sample={'性别' '身高' 'Nan';'女' 1.60 '矮'; '男' 2.00 '高'; '女' 1.90 '中';
    '女'  1.88  '中';  '女'  1.70  '矮';  '男' 1.85  '中'; '女'  1.60  '矮';
    '男'  1.70  '矮';  '男'  2.20  '高';  '男' 2.10  '高'; '女'  1.80  '中';
    '男'  1.95  '中';  '女'  1.90  '中';  '女' 1.80  '中'; '女'  1.75  '中'};
>>rule=mytree_decisionCART(sample,[0 1],0.9);
```

根据结果，可知：不能形成决策规则，请增加属性或降低正确率。

从题意中明显可看出，此例的属性太少。如果将正确率降为0.8，则可以形成决策规则：

```
 rule{1} = '身高'   '<1.8'   '矮'
```

例2.6　在建立有效的决策树时，不仅需要考虑分类的准确性，而且需要考虑决策树的复杂程度，即建立的决策树，在保证具有一定的分类正确率条件下，越简化越好。最常用的决策树简化方法就是剪枝。下面利用后剪枝技术对例2.1中形成的决策树进行剪枝，其中所用的测试样本集见表2.5。

表 2.5　测试样本数据

属　性	Outlook	Temperature	Humidity	Windy	类　别
1	Overcast	Hot	Normal	Not	P
2	Overcast	Mild	High	Very	N

续表

属　性	Outlook	Temperature	Humidity	Windy	类　别
3	Overcast	Cool	Normal	Medium	P
4	Overcast	Hot	High	Not	P
5	Sunny	Hot	Normal	Medium	P
6	Sunny	Hot	High	Not	P
7	Sunny	Hot	High	Medium	N
8	Sunny	Mild	Normal	Not	P
9	Rain	Cool	High	Medium	N
10	Rain	Hot	Normal	Very	N
11	Rain	Mild	High	Very	N
12	Rain	Cool	High	Medium	N

解：

根据决策树各种算法的原理，再结合决策树剪枝方法，可进行编程计算，限于篇幅，不再详细列出此程序。此例中的程序较为完整，既可以处理数值属性，也可以处理其他属性以及具有混合属性训练样本集。

```
>>train_sample={'outlook' 'temperature' 'humidity' 'windy' 'Nan'
'overcast' 'hot' 'high' 'not' 'N'; 'overcast' 'hot' 'high' 'very' 'N';'overcast'
'hot' 'high' 'medium' 'N'
'sunny' 'hot' 'high' 'not' 'P';'sunny' 'hot' 'high' 'medium' 'P';'rain' 'mild'
'high' 'not' 'N'
'rain' 'mild' 'high' 'medium' 'N';'rain' 'hot' 'normal' 'not' 'P';'rain' 'cool'
'normal' 'medium' 'N'
'rain' 'hot' 'normal' 'very' 'N';'sunny' 'cool' 'normal' 'very' 'P';'sunny' 'cool'
'normal' 'medium' 'P'
'overcast' 'mild' 'high' 'not' 'N';'overcast' 'mild' 'high' 'medium'
'N';'overcast' 'cool' 'normal' 'not' 'P'
'overcast' 'cool' 'normal' 'medium' 'P';'rain' 'mild' 'normal' 'not' 'N';'rain'
'mild' 'normal' 'medium' 'N'
'overcast' 'mild' 'normal' 'medium' 'P';'overcast' 'mild' 'normal' 'very'
'P';'sunny' 'mild' 'high' 'very' 'P'
'sunny' 'mild' 'high' 'medium' 'P';'sunny' 'hot' 'normal' 'not' 'P';'rain' 'mild'
'high' 'very' 'N'};
>>test={'outlook' 'temperature' 'humidity' 'windy' 'Nan'
'overcast' 'hot' 'normal' 'not' 'P';'overcast' 'Mild' 'high' 'very'
'N';'overcast' 'cool' 'normal' 'medium' 'P'
'overcast' 'hot' 'high' 'not' 'P';'sunny' 'hot' 'normal' 'medium' 'P';'sunny'
```

```
'hot' 'high' 'not' 'P'
'sunny' 'hot' 'high' 'medium' 'N';'sunny' 'Mild' 'normal' 'not' 'P';'rain' 'cool'
'high' 'medium' 'N'
'rain' 'hot' 'normal' 'very' 'N';'rain' 'Mild' 'high' 'very' 'N';'rain' 'cool'
'high' 'medium' 'N'};
>>rule=mytree_decisionID3_1(train_sample,test);        %ID3算法函数，包括剪枝处理
>> rule={1x3 cell} {1x5 cell} {1x5 cell} {1x3 cell}    %得到决策规则
>>rule{1}= 'outlook'    'sunny'         'P'
>>rule{2}= 'outlook'    'overcast'      'humidity'    'high'     'N'
>>rule{3}= 'outlook'    'overcast'      'humidity'    'normal'   'P'
>>rule{4}= 'outlook'    'rain'          'N'
```

读书笔记

第 3 章

人工神经网络算法

3.1　人工神经网络概述

人工神经网络（Artificial Neural Network，ANN）有时简称为神经网络，是在现代生物学研究人脑组织所取得成果的基础上提出的，它利用大量简单的处理单元广泛连接组成的复杂网络，来模拟人类大脑的神经网络结构和行为。它的研究成果显示了人工神经网络具有人脑功能的基本物质特征——学习、记忆、概括、归纳和抽取等，从而解决了人工智能研究中的某些局限性。它不同于以前人工智能领域中普遍采用的基于逻辑和符号处理的理论和方法，而是开辟了崭新的途径。

神经网络的产生是从生物学上获得的灵感，它将实现模拟生物神经元的某些基本功能的元件组织起来，而组织方式或是按人脑组织方式，或是根本不按人脑组织方式，不管其是高度相似，还是简单模仿，神经网络仍能显示其惊人的与人脑相近的特性。例如，它能学习专门知识，从先前已有的实例中概括出新的例子。

随着神经网络的大量开创性应用，可以发现，不管网络的组织形式如何，它们均有以下三个共同的特性。

（1）学习。神经网络具有对周围环境自学习、自适应的功能。这种学习机制基于网络的组织形式能适应各种学习算法，而学习算法是指网络能根据训练实例来决定自身的行为，当出现一组输入信息（或附有所需的输出结果）时，它们能不断调整，产生一系列一致的结果，犹如人们智能活动“习惯成自然”一样，反映出网络的学习性能。

（2）概括。一旦训练后，神经网络的响应能在某种程度上对外界输出信息的少量丢失或神经网络组织的局部缺损不再很敏感。这种机制与大脑每日有大量神经网络正常死亡但并不影响大脑的功能，或者大脑局部损伤会引起某些功能的逐渐衰退，但不是功能完全丧失一样，反映了神经网络的鲁棒性，即具有容错能力。

（3）抽取。神经网络还有一种抽取外界输入信息特征的特殊功能，可以从不完善的数据和图形进行学习和做出决定。一旦训练完成，就能从给定的输入模式快速计算出结果。如对它进行一张人像的一系列不完整的照片识别训练之后，再任选一张缺损的照片让神经网络识别，网络将会做出一个完整形式的人像照片的响应。在某种意义上可以说它能“创造”出以前从未见到的某些东西。

人工神经网络的这些基本特征反映了它能较之其他人工智能理论和方法更好地解决这方面的任务。同时，也可以看出它实现的是右半脑直觉形象思维的特性，而传统人工智能理论和方法实现左半脑逻辑思维的特性，它们有着互补的作用，而不是简单取代的关系。正是具有这些特点，人工神经网络在人工智能、自动控制、计算机科学、信息处理、模式识别等领域得到了广泛的应用。

3.2　人工神经网络的基本模型

人工神经网络系统是大脑生物结构的数学模型，由大量功能简单而具有自适应能力的信息处理单元即人工神经元按照大规模并行的方式，通过拓扑结构连接而成。

3.2.1　神经元

人工神经元是对生物神经元的模拟。在生物神经元上，来自轴突的输入信号神经元终结于突触上。信息是沿着树突传输并发送到另一个神经元；对于人工神经元，这种信号传输由输入信号

x、突触权重 w、内部阈值 θ_j 和输出信号 y 来模拟，如图 3.1 所示。

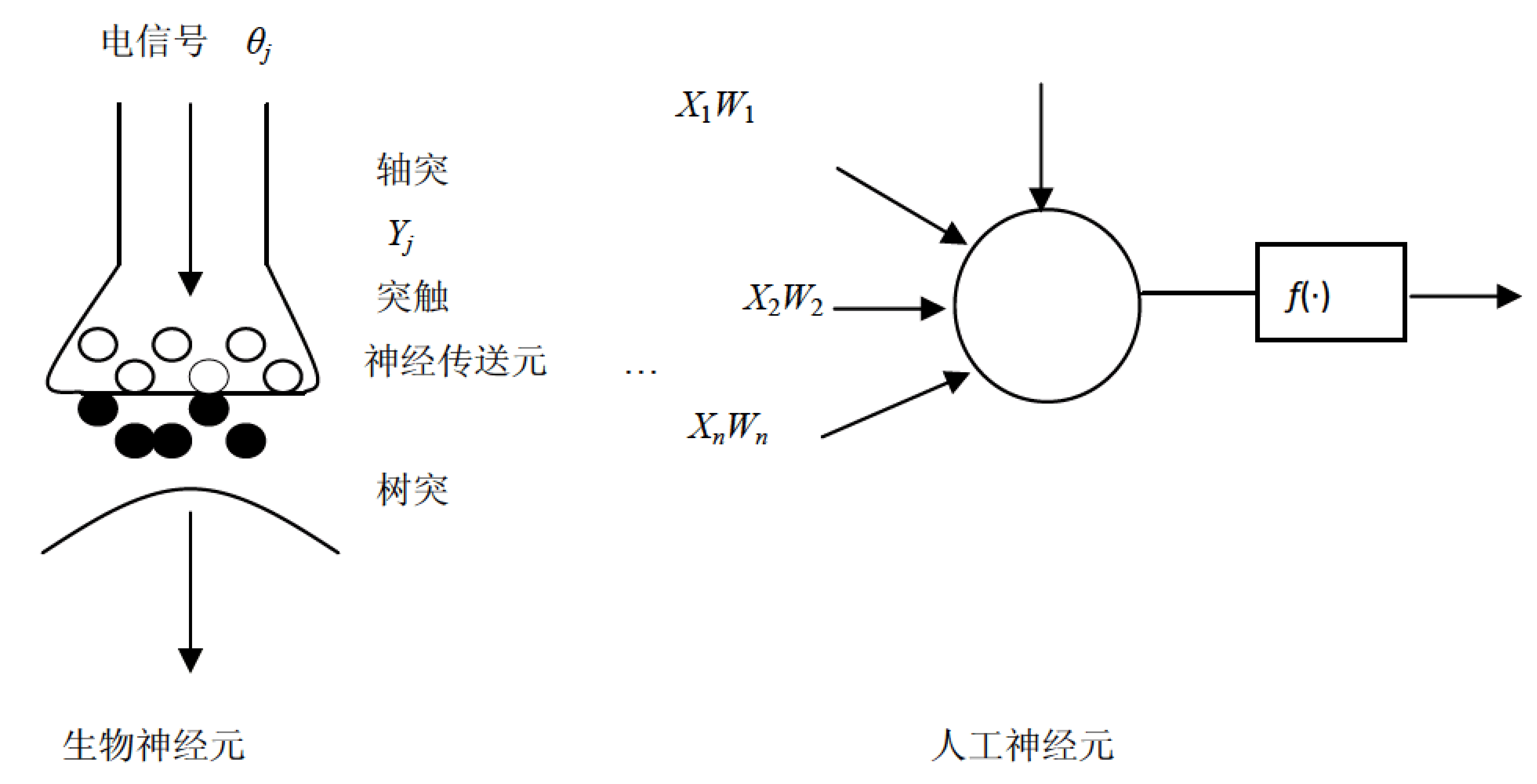


图 3.1　生物和人工神经元结构示意图

3.2.2　传递函数

在人工神经元系统中，其输出是通过传递函数 $f(\cdot)$ 来完成的。传递函数的作用是控制输入对输出的激活作用，把可能的无限域变换到给定范围的输出，对输入、输出进行函数转换，以模拟生物神经元线性或非线性转移特性。

由图 3.1 可见，简单神经元主要由权值、阈值和 $f(\cdot)$ 的形式定义，其数学表达式如下：

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i - \theta_i\right)$$

可以选择传递函数为所希望的函数形式，如平方根、乘积、log、e^x 等，表 3.1 为一些常用的传递函数。除线性传递函数外，其他变换给出的均是累积信号的非线性变换。因此，人工神经网络特别适合于解决非线性问题。

表 3.1　神经网络传递函数

类　型	函　数
阈值逻辑（二值）	$f(x)=\begin{cases}1(x\geqslant s)\\0(x<s)\end{cases}$
阈值逻辑（两极）	$f(x)=\begin{cases}1(x\geqslant s)\\-1(x<s)\end{cases}$
线性传递函数	$f(x)=c\cdot x$
线性阈值函数	$f(x)=\begin{cases}1(x\geqslant s)\\0(x<s)\\c(\text{其他})\end{cases}$
Sigmoid 函数	$f(x)=\dfrac{1}{1+e^{-c\cdot x}}$
双曲线－正切函数	$f(x)=\dfrac{e^{cx}-e^{-cx}}{e^{cx}+e^{-cx}}$

3.2.3 人工神经网络的分类

人工神经网络模型可以有多种形式，它取决于网络的拓扑结构、神经元传递函数、学习算法和系统特点。一般可根据以下方式进行分类。

- 按结构方式分类，有前馈网络和反馈网络，如 BP 前馈网络和反馈 Hopfield 网络。
- 按状态方式分类，有离散型网络和连续型网络，如 Hopfield 离散型网络和 Hopfield 连续型网络。
- 按学习方式分类，有监督学习网络和无监督学习网络，如 BP、RBF 等有学习监督网络和 Kohonet 无监督学习网络。

3.3 BP 神经网络

1985 年，Rumelhart 提出的 Error back propagation 算法（简称 BP 算法），系统地解决了多层网络中隐单元层连接权的学习问题。目前 BP 模型已成为人工神经网络的重要模型之一，并得到了广泛的应用。

3.3.1 BP 人工神经网络结构

BP 人工神经网络由输入层、隐含层和输出层三层组成，其核心是通过一边向后传递误差，一边修正误差的方法来不断调节网络参数（权、阈值），以实现或逼近所希望的输入、输出映射关系。BP 人工神经网络结构如图 3.2 所示。

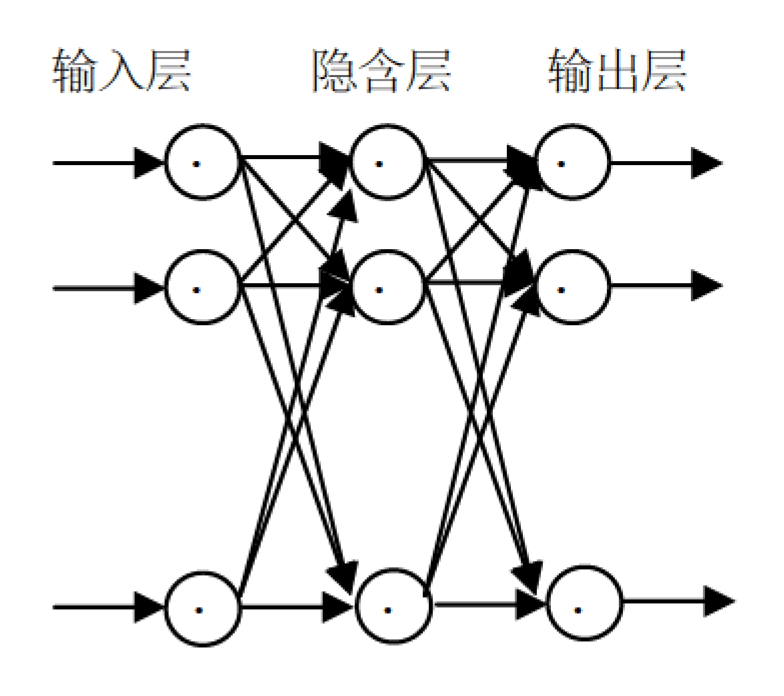


图 3.2 BP 人工神经网络结构

3.3.2 BP 人工神经网络的学习算法

BP 人工神经网络的学习算法，包含以下 6 步：

（1）初始化。为了加快网络的学习效率，一般需对原始数据的输入、输出样本进行规范化处理；给权值及阈值赋予（−1,1）区间的随机值。

（2）进入循环。计算网络的输入和输出值。

隐含层各节点的输入、输出分别为

$$s_j^k=\sum_{i=1}^{n}a_i^k w_{ij}-\theta_j \qquad b_j^k=\frac{1}{1+e^{-s_j^k}}\text{，}j=1,2,\cdots,p\text{（隐含层单元数）}$$

输出层各节点的输入、输出分别为

$$l_t^k=\sum_{j=1}^{p}b_j^k v_{jt}\qquad c_t^k=\frac{1}{1+e^{-l_t^k}}\text{，}t=1,2,\cdots,q\text{（输出层神经元数）}$$

（3）误差逆传播。各连接层及阈值的调整，按梯度下降法的原则进行。

设网络的计算输出为 c_t^k，则网络的希望输出 y_t^k 与计算输出 c_t^k 的偏差的均方值 E_k

$$E_k=\sum_{t=1}^{q}\frac{(y_t^k-c_t^k)^2}{2}$$

计算输出层各节点的误差 d_t^k

$$d_t^k=(y_t^k-c_t^k)c_t^k(1-c_t^k),t=1,2,\cdots,q$$

隐含层各节点的误差 h_j^k

$$h_j^k=\left[\sum_{t=1}^{q}d_t^k v_{jt}\right]b_j^k(1-b_j^k),j=1,2,\cdots,q$$

（4）修正权值、阈值。用输出层、隐含层各节点的误差修正各层的连接权值及阈值。

$$v_{jt}(N+1)=v_{jt}(N)+\alpha d_t^k b_j^k$$
$$\gamma_t(N+1)=\gamma_t(N)-\alpha d_t^k$$
$$w_{ij}(N+1)=w_{ij}(N)+\beta h_j^k a_i^k$$
$$\theta_j(N+1)=\theta_j(N)-\beta h_j^k$$

其中：N 为修正次数。

以上循环执行 m 次。

（5）若网络的全局误差小于指定的值，则算法转入第（6）步，否则转入第（2）步。

（6）计算输出层。

3.4　RBF 神经网络

RBF 网络是 20 世纪 80 年代提出的一种人工神经网络结构，是具有单隐层的前向网络。它不仅可以用来函数逼近，也可以进行预测。

3.4.1　RBF 网络结构

RBF 网络由两层组成，第一层为隐含的径向基层，第二层为输出线性层，其网络结构如图 3.3 所示。

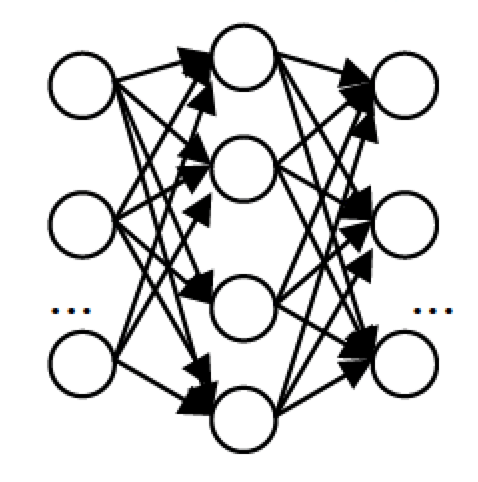


图 3.3　RBF 网络结构

从图 3.3 中可看出，RBF 网络的输入层实现从 $x \to R_i(x)$的非线性映射，输出层实现从 $R_i(x) \to y_k$ 的线性映射，即

$$y_k = \sum_{i=1}^{p} w_{ki} R_i(x),\ k=1,2,\cdots,q$$

其中：q 是输出节点数；w_{ki} 为输出层第 k 个神经元与隐含层第 j 个神经元之间的调节权重。

从理论上讲，RBF 人工神经网络可以逼近任何的非线性函数。

RBF 人工神经网络中径向基函数是径向对称的，最常用的是高斯函数

$$R_i(x) = \exp\left(-\frac{\| x - c_i \|^2}{2\sigma_i^2}\right),\ i=1,2,\cdots,p$$

其中：x 是 m 维输入向量；c_i 是第 i 个基函数的中心；σ_i 是第 i 个感知的变量；p 是感知单元的个数；$\|x-c_i\|^2$ 是向量 $x-c_i$ 的范数。

3.4.2 RBF 人工神经网络的学习算法

RBF 人工神经网络的学习算法包含以下几步。

（1）初始化。对连接权重 w、各神经元的中心参数 c、宽度向量 σ 等参数的按一定的方式进行初始化，并给定 α（调节系数）和 η（学习因子）的取值。

（2）计算隐含层的输出。利用高斯函数计算隐含层的输出。

（3）计算输出层神经元的输出。利用下式求出输出神经元的输出

$$y_k = \sum_{i=1}^{p} w_{ki} R_i(x)$$

（4）误差调整。对各初始化值，根据下列公式进行迭代计算，以自适应调节到最佳值。

$$w_{kj}(t) = w_{kj}(t-1) - \eta \frac{\partial E}{\partial w_{kj}(t-1)} + \alpha[w_{kj}(t-1) - w_{kj}(t-2)]$$

$$c_{ji}(t) = c_{ji}(t-1) - \eta \frac{\partial E}{\partial c_{ji}(t-1)} + \alpha[c_{ji}(t-1) - c_{ji}(t-2)]$$

$$\sigma_{ji}(t) = \sigma_{ji}(t-1) - \eta \frac{\partial E}{\partial \sigma_{ji}(t-1)} + \alpha[\sigma_{ji}(t-1) - \sigma_{ji}(t-2)]$$

其中：$w_{kj}(t)$为第 k 个输出神经元与第 j 个隐含层神经元之间有第 t 次的迭代计算时的调节权重；$c_{ji}(t)$为第 j 个隐含层对应于第 i 个输入神经元在第 t 次迭代计算时的中心分量；$\sigma_{ji}(t)$为与中心 $c_{ji}(t)$ 对应的宽度；η 为学习因子；E 为 RBF 神经网络误差函数，由下式给出

$$E = \frac{1}{2}\sum_{l=1}^{N}\sum_{k=1}^{q}(y_{lk} - O_{lk})^2$$

其中：O_{lk} 为第 k 个输出神经元在第 l 个输入样本时的期望输出值；y_{lk} 为第 k 个输出神经元在第 l 个输入样本时的网络输出值。

（5）按下式计算网络输出的均方根误差 RMS 的值，若 RMS≤ε，则训练结束，计算输出，否则转到第（2）步。

$$\mathrm{RSE}=\sqrt{\frac{\sum_{l=1}^{N}\sum_{k=1}^{q}(y_{lk}-O_{lk})^2}{qN}}$$

3.5　SOM 神经网络

自组织特征映射网络（Self-organizing Feature Map）也称 Kohonen 网络，它是一个由全连接的神经元阵列组成的无教师、自组织、自学习网络。

3.5.1　SOM 神经网络结构

SOM 网络一般只包含一维阵列和二维阵列，典型的二维阵列 SOM 神经网络结构如图 3.4 所示，由输入层和竞争层（或称映射层）组成。输入层神经单元数为 m，竞争层由 $a\times b$ 个神经元组成的二维平面阵列，输入层与竞争层各神经之间实现全连接。

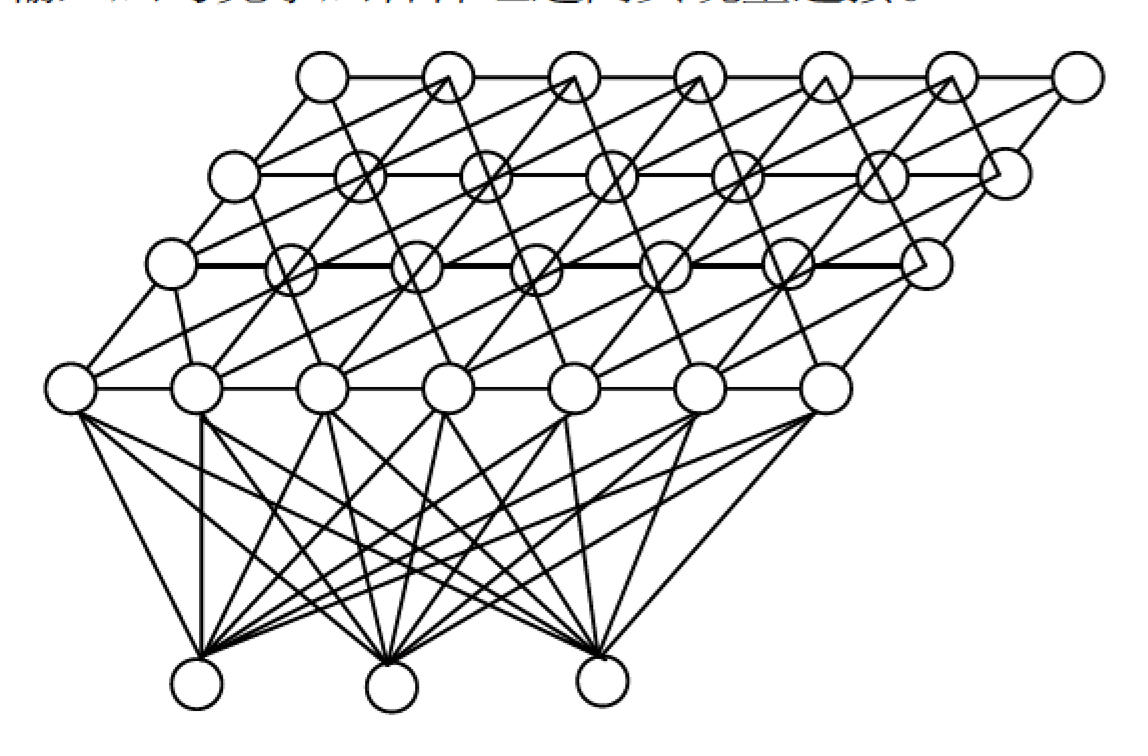


图 3.4　二维阵列 SOM 神经网络结构模型

3.5.2　SOM 神经网络学习算法

SOM 神经网络能够自动找出输入数据之间的类似度，将相似的输入在网络上就近配置，就可以构成对输入数据有选择地给予反应。其学习算法如下。

（1）网络初始化。用随机数设定输入层和映射层之间的初始值。对 m 个输入神经元到输出神经元的连接权值赋予较小的权值。选取输出神经元 j 个“邻接神经元”的集合 S_j，其中，$S_j(0)$ 表示时刻 t=0 的神经元 j 的“邻接神经元”的集合，$S_j(t)$表示时刻 t 的“邻接神经元”的集合，区域 $S_j(t)$随着时间的增长而不断缩小。

（2）输入向量的输入。将输入向量 $X=(x_1,x_2,\cdots,x_m)$输入给输入层神经单元。

（3）计算映射层的权值向量和输入向量的距离（欧式距离）。在映射层，按下式计算各神经元的权值向量和输入向量的欧式距离

$$d_j=\|X-W_j\|=\sqrt{\sum_{i=1}^{m}(x_i(t)-w_{ij}(t))^2}$$

式中：w_{ij} 为输入层的 i 神经元和映射层的 j 神经元之间的权值。通过计算得到一个具有最小距离的神经元，即胜出神经元，记为 j^*。

（4）权值的学习。按下式修正输出神经元 j^* 及其“邻接神经元”的权值。

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(t)(x_i(t) - w_{ij}(t))$$

式中：η 为[0,1]区间内的一个常数，随着时间变化逐渐下降到 0。

$$\eta(t) = \frac{1}{t} \text{或} \eta(t) = 0.2\left(1 - \frac{t}{10\ 000}\right)$$

（5）计算输出 O_k。按下式计算输入

$$O_k = f(\min_j \|X - w_{ij}\|)$$

其中：$f(*)$一般为 0～1 函数或者其他非线性函数。

（6）是否达到预先设定的要求。如达到要求则算法结束；否则，返回步骤（2），进入下一轮学习。

3.6 反馈型神经网络（Hopfield）

Hopfield 网络是最典型的反馈网络模型，是目前人们研究最多的模型之一。它由相同的神经网络元构成的单层，并且具有学习功能的自联想网络，可以完成制约优化和联想记忆等功能。

3.6.1 Hopfield 网络的拓扑结构

Hopfield 网络的拓扑结构如图 3.5，其中第一层仅是作为网络的输入，它不是实际的神经元，没有计算功能。第二层是实际神经元，执行对输入信息与系数的乘积求累加和，并经非线性函数 f 处理后产生输出信息。f 是一个简单的阈值函数，如果神经元的输出信息大于阈值 θ，那么神经元的输出就取值为 1，小于阈值 θ，则神经元的输出就取值为−1。

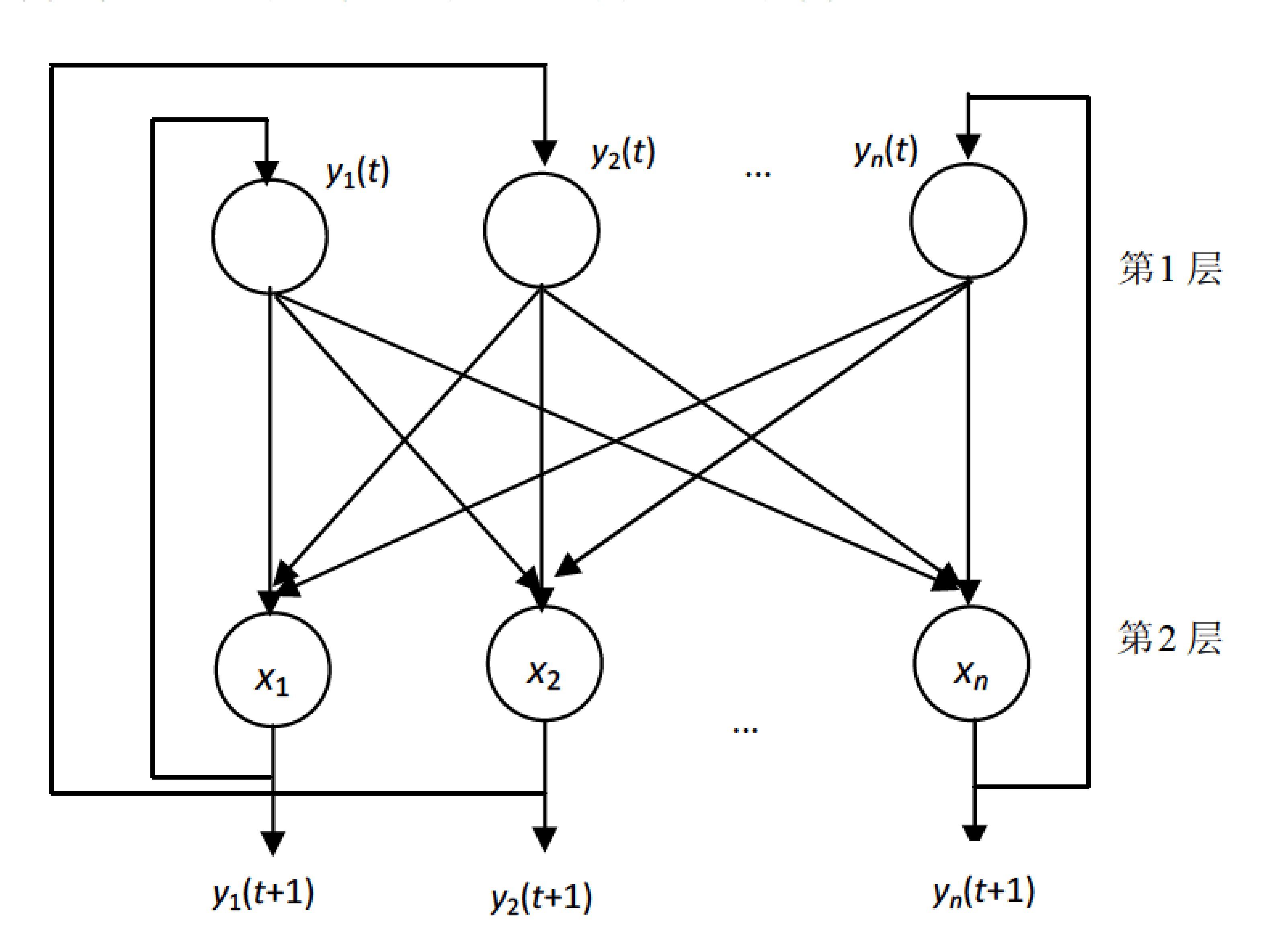


图 3.5 Hopfield 网络拓扑结构

从图 3.5 中可看出，Hopfield 网络是一种循环神经网络，由于其输出端有反馈到其输入端，

所以 Hopfield 网络在输入的激励下，会不断产生状态变化。当有输入之后，可以求得 Hopfield 的输出。这个输出反馈到输入从而产生新的输出，这个反馈过程一直进行下去。如果 Hopfield 网络是一个能收敛的稳定网络，则这个反馈和迭代的计算过程所产生的变化越来越小。一旦到达了稳定平衡状态，那么 Hopfield 网络就会输出一个稳定的恒值。

3.6.2　Hopfield 网络的学习算法

Hopfield 网络的训练和分类利用的是 Hopfield 网络的联想记忆功能。当它做联想记忆时，首先通过一个学习训练过程确定网络中的权重，使所记忆的信息在网络的 n 维超立方体的某一个顶角的能量最小。当网络的权值被确定之后，只要向网络给出输入向量，即使这个向量是不完全或部分不正确的数据，但网络仍然产生所记忆的信息的完整输出。

Hopfield 网络的学习算法如下。

（1）确定参数。将输入向量 X，即 $X=[x_{i1},x_{i2},\cdots,x_{in}]^{\mathrm{T}}$ 存入 Hopfield 网络中，则在网络中第 i,j 两个节点间的权重系数按下列公式计算

$$w_{ij}=\begin{cases}\sum_{k=1}^{N}x_{ki}x_{kj} & i\neq j\\ 0 & i=j\end{cases},i\ j=1,2,\cdots,n$$

确定输出向量 $Y=[y_1,y_2,\cdots y_n]^{\mathrm{T}}$。

（2）对等测样本进行分类。对于待测样本，通过对 Hopfield 网络构成的联想存储器进行联想检索过程实现分类：

① 将 X 中各个分量的 $x_1,x_2,\cdots,x_n$ 分别作为第一层网络 n 个节点的输入，则节点有相应的初始状态 Y（t=0），即 $y_i(0)=x_j$，j=1,2,$\cdots$,n。

② 对于二值神经元，计算当前 Hopfield 网络输出

$$U_j(t+1)=\sum_{i=1}^{n}w_{ij}y_i(t)+x_j-\theta_j,j=1,2,\cdots,n$$

$$y_i(t+1)=f(U_j(t+1)),j=1,2,\cdots,n$$

式中：x_j 为外部输入；f 是非线性函数，可以选择阶跃函数；θ_j 为阈值函数。

$$f(U_j(t+1))=\begin{cases}-1 & U_j(t+1)<0\\ 1 & U_j(t+1)\geqslant 0\end{cases}$$

③ 对于一个网络来说，稳定性是一个重要的性能指标。对于离散的 Hopfield 网络，其状态为 $Y(t)$，如果对于任何 $\Delta t>0$，当网络从 t=0 开始，有初始状态 $Y(0)$，经过有限时间 t，有 $Y(t+\Delta t)=Y(t)$，则称网络是稳定的，此时的状态称为稳定状态。通过网络状态不断变化，最后状态会稳定下来，最终的状态是与待测样本向量 X 最接近的训练样本。所以，Hopfield 网络的最终输出，也就是待测样本向量联想的检索结果。

④ 利用最终输出与训练样本进行匹配，找出最相近的训练样本向量，其类别即是等测样本类别。所以，即使待测样本并不完全或部分不正确，也能找到正确的结果。

3.7 基于 MATLAB 的神经网络方法

人工神经网络在故障诊断、特征的提取和预测、非线性系统的自适应控制、不能用规则或公式描述的大量原始数据的处理等方面具有比经典计算方法优越的性能，且有极大的灵活性和自适应性。

在实际应用中，面对一个实际问题，如要用人工神经网络求解，首先应根据问题的特点，确定网络模型，再通过网络仿真分析，分析确定网络是否适合实际问题的特点。在此过程中，应注意以下几个问题。

3.7.1 信息表达方式

各种应用领域的信息有不同的物理意义和表示方法，为此要将这些不同物理意义和表示方法的信息转化为网络所能表达并能处理的形式。不同应用领域的各种数据形式一般为以下几种。

（1）已知数据样本；

（2）已知一些相互关系不明的数据样本；

（3）输入—输出模式为连续量、离散量；

（4）具有平移、旋转、伸缩等变化的模式。

3.7.2 网络模型选择

也即确定激活函数、连接方式、各神经元的相互作用等；当然也可以针对问题的特点，对原始网络模型进行变形、扩充等处理。

3.7.3 网络参数选择

确定输入、输出神经元的数目、多层网的层数和隐含层神经元的数目等。

3.7.4 学习训练算法选择

确定网络学习时的学习规则及改进学习规则。在训练时，还要结合实际问题考虑网络的初始化。

3.7.5 系统仿真的性能对比

将应用神经网络解决的领域问题与其他采用不同方法的仿真系统的效果进行比较，以检验方法的准确度和解决问题的精度。

例 2.7 蠓虫分类问题可概括叙述如下：生物学家试图对两种蠓虫（Af 与 Apf）进行鉴别，依据的资料是触角和翅膀的长度，已经测得了 9 只 Af 和 6 只 Apf 的数据，如表 3.2 所示。

表 3.2 样本数据集

触 角	长 度	类 别	触 角	长 度	类 别
1.24	1.27	Af	1.14	1.82	Apf
1.38	1.64	Af	1.18	1.96	Apf
1.38	1.82	Af	1.20	1.86	Apf

续表

触　角	长　度	类　别	触　角	长　度	类　别
1.38	1.90	Af	1.26	2.00	Apf
1.40	1.70	Af	1.28	2.00	Apf
1.48	1.82	Af	1.30	1.96	Apf
1.54	1.82	Af			
1.56	2.08	Af			
1.36	1.74	Af			

根据以上资料，求解下列问题。

（1）根据如上资料，如何制定一种方法，正确地区分两类蠓虫。

（2）对触角和翼长分别为(1.24,1.80)、(1.28,1.84)与(1.40,2.04)的三个标本，用所得到的方法加以识别。

（3）设 Af 是宝贵的传粉益虫、Apf 是某疾病的载体，是否应该修改分类方法。

解：

在 MATLAB 中，利用人工神经网络解决各种实际问题，一般有两种方法：一是利用命令行；二是利用神经网络图形用户界面（GUI）。

对于此题，利用 BP 神经网络的命令行进行求解：

```
>>clear
p1=[1.24,1.27;1.36,1.74;1.38,1.64;1.38,1.82;1.38,1.90;1.40,1.70;1.48,1.82;
1.54,1.82;1.56,2.08];
p2=[1.14,1.82;1.18,1.96;1.20,1.86;1.26,2.00;1.28,2.00;1.30,1.96];
p=[p1;p2]';pr=minmax(p);goal=[ones(1,9),zeros(1,6);zeros(1,9),ones(1,6)];
plot(p1(:,1),p1(:,2),'h',p2(:,1),p2(:,2),'o')
net=newff(pr,[3,2],{'logsig','logsig'});
net.trainParam.show = 10;net.trainParam.lr = 0.05;net.trainParam.goal = 1e-10;
net.trainParam.epochs = 50000;net = train(net,p,goal);
x=[1.24 1.80;1.28 1.84;1.40 2.04]';
y0=sim(net,p);
y=sim(net,x);              %实际样本的分类结果
```

求得结果如下，从结果可看出，实际样本基本上属于第二类即为Apf。

```
    y = 0.0002   0.0120      0.1437
         0.9996      0.9840      0.8286
```

样本的分布图如图3.6所示。

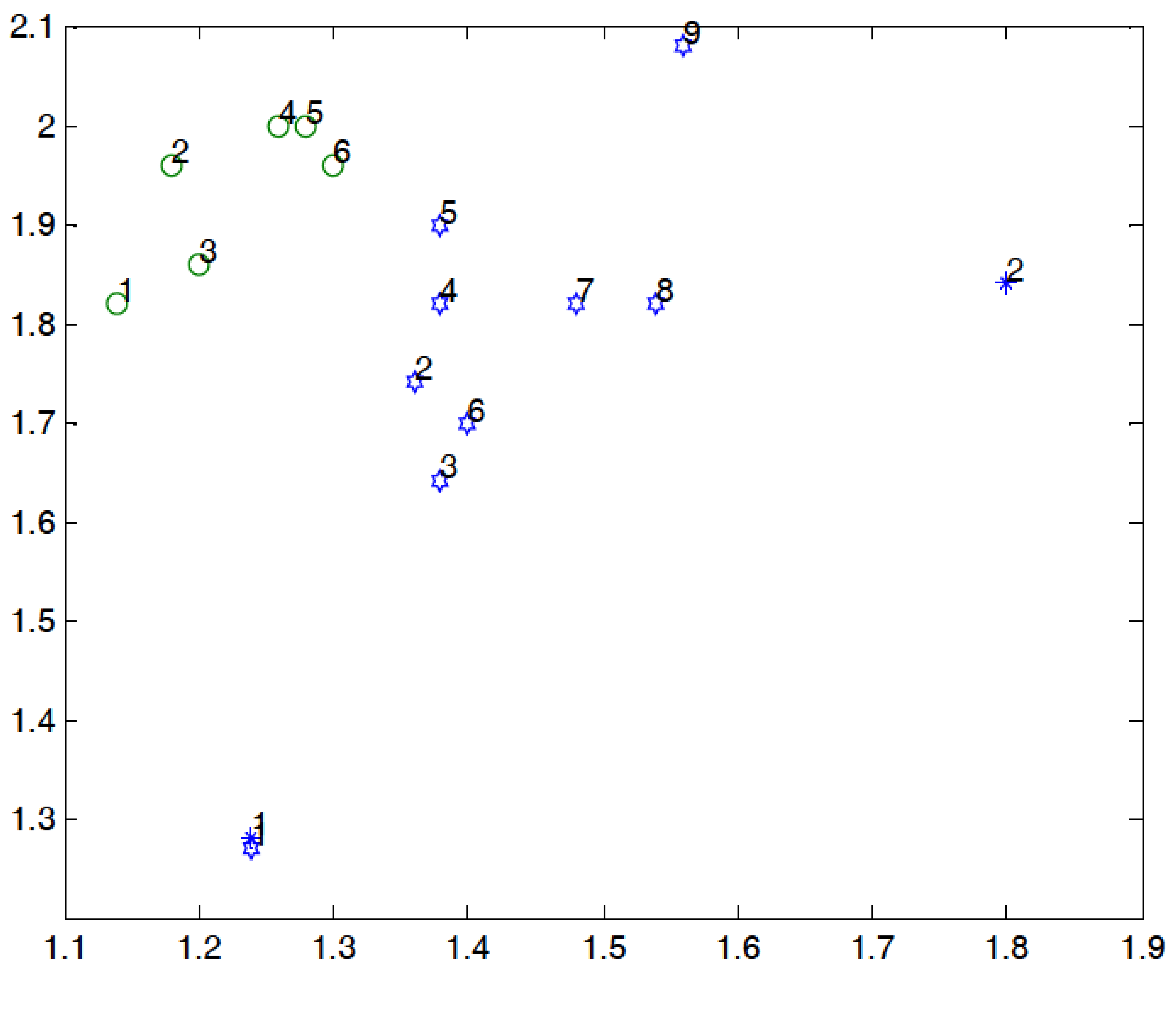


图 3.6　样本数据分布图

例2.8　利用神经网络方法对表3.3的数据进行分析。

表 3.3　煤样各指标的实测数据

煤样样本分类及编号	特性指标									
	碳	氢	硫	氧	镜质组分	丝质组分	块状微粒体	粒状微粒体	壳质树脂体	平均最大反射率
无烟煤	92.21	2.74	0.84	3.58	86.70	13.30	0.00	0.00	0.00	4.92
	92.58	2.80	1.00	2.98	90.01	9.70	0.20	0.00	0.00	3.98
	92.63	3.04	0.74	2.64	89.10	0.60	0.30	0.00	0.00	4.12
	93.01	1.98	0.55	3.46	89.00	9.40	0.80	0.00	0.00	6.05
	93.01	2.79	0.79	2.67	88.30	11.70	0.00	0.00	0.00	4.50
烟煤	84.62	5.61	0.76	7.30	69.10	13.10	1.40	4.10	12.50	0.90
	84.53	5.55	0.70	7.36	64.60	8.10	3.00	11.3	11.00	0.85
	83.82	5.78	0.90	7.80	84.10	2.70	1.20	7.40	4.50	0.93
	82.65	5.57	2.48	7.19	77.20	9.10	2.70	3.20	7.80	0.83
	82.43	5.77	1.61	8.53	84.90	3.80	2.30	5.00	4.10	0.84
	81.88	5.87	2.94	7.39	80.30	4.30	3.30	7.80	4.30	0.71
褐煤	72.49	5.31	2.11	20.23	85.72	7.90	3.54	3.12	3.73	0.30
	72.29	5.26	1.02	20.43	85.60	4.60	3.30	2.80	3.70	0.31
	71.39	5.33	1.07	21.03	84.70	5.90	2.80	3.00	3.60	0.32
	70.95	5.04	1.50	21.10	81.85	7.25	2.75	2.94	3.21	0.33
	71.85	5.17	1.14	20.95	85.10	7.21	3.54	2.77	3.54	0.32

解：

对于此例，利用神经网络图形用户界面（GUI）进行分析。

首先在工作空间输入数据，即属性数据 *x* 和目标分类数据 *y*，然后打开神经网络模式识别工具箱图形界面：

```
>>nprtool
```

出现如图 3.7 所示的图形。

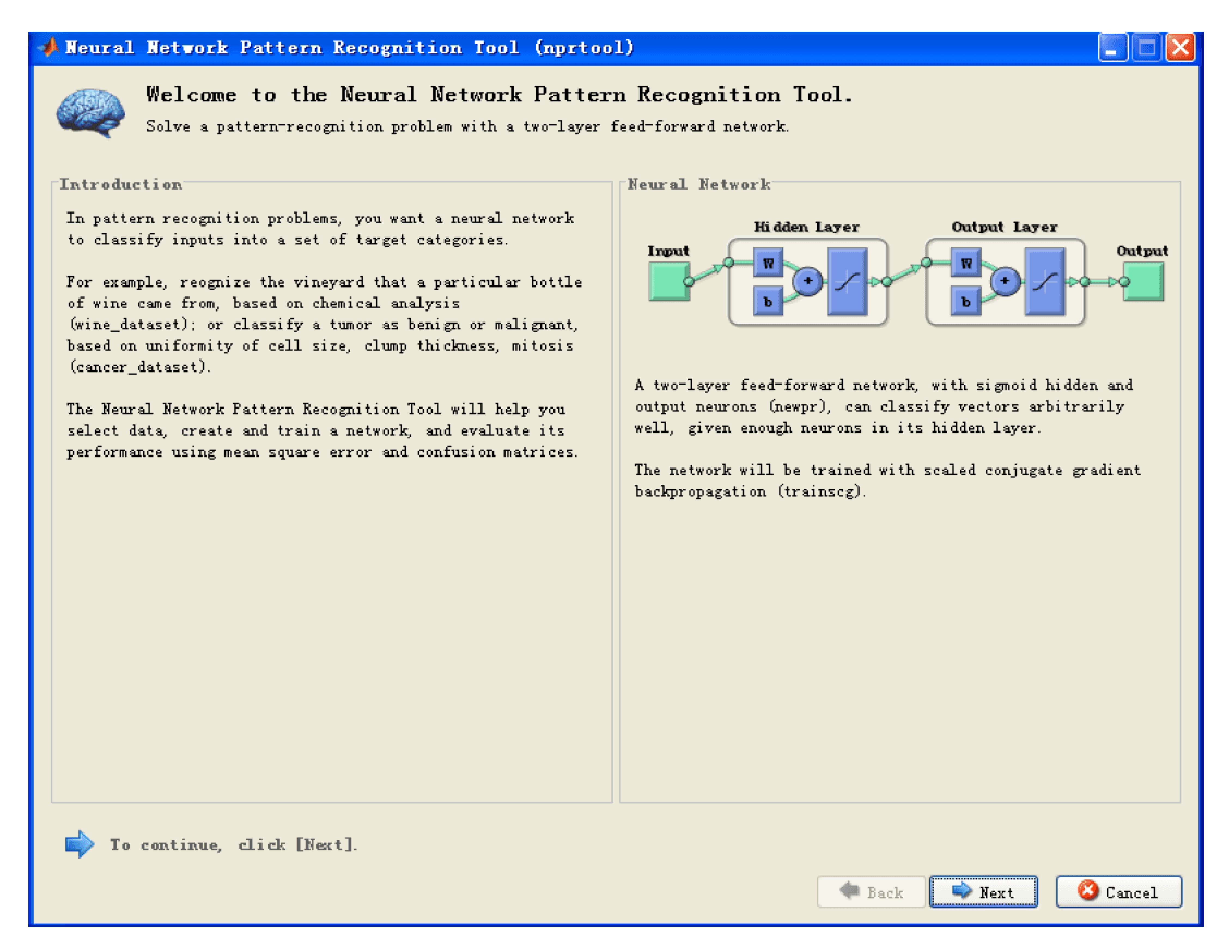


图 3.7　神经网络模式识别用户图形界面

单击图形中的 Next，进入数据导入，选择从工作空间中输入数据，分别导入输入数据(x)及目标数据(y)。要注意目标数值应是二值类型，对于本例为：y=[0 0 0 0 0 0 0 0 0 0 0 1 1 1 1;0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]，分别代表 1、2 和 3 类。并且在输入数据和目标数据的格式相同时，才能进入验证和测试样本。在此对话框对输入的样本进行训练、验证和测试样本选择，可以选择不同的比例。然后单击 Next，进入网络选择对话框，在此设置隐含层数目，默认为 20。

单击 Next，进入网络训练对话框。单击 Retrain，便可以对网络进行训练和查看训练结果。单击 Next，进入网络评估对话框，在此可以再训练或进入下一对话框进行网络保存。以相应名字将结果输入到命令窗口，便可以查看分类结果及对未知样本进行预测。对于本例为：

```
output1=[0.0001  0.0001  0.0001 0.0004  0.0001  0.0005  0.0007  0.0005  0.0006
         0.0005  0.0004  0.0004 0.0004  0.0003  0.9997  1.0000  0.9993  0.9992]
```

```
output2=[0.0142  0.0008  0.9996  0.9998  0.9998  0.9996  0.9999
         0.9997  0.9998  0.9996  0.9999  0.9999  0.9999  0.9998]
```

例 2.9　给定待拟合的曲线形式为：$f(x) = 0.5 + 0.4\sin(2\pi x)$。

在 $f(x)$上等间隔取 30 个点的数据，在此数据的输出值上加均值为 0，均方差 σ=0.05 的正态分布噪声作为给定训练数据，用多项式拟合此函数，分别取多项式的阶次为 1、3 和 11 阶，图示出拟合结果，并讨论多项式阶次对拟合结果的影响。

解：

```
>> x=linspace(-6,6,30);for
i=1:length(x);y(i)=0.5+0.4*sin(2*pi*x(i))+normrnd(0,0.05);end
P1=polyfit(x,y,1);y1=polyval(P1,x);P2=polyfit(x,y,3);y2=polyval(P2,x);P3=poly
fit(x,y,11);
y3=polyval(P3,x);
>>nftool
```

打开如图3.8神经网络拟合图形用户对话框，然后单击Next，进入Select Data对话框，从工作空间中分别选择数据（x,y1）、（x,y2）、（x,y3），对1阶、3阶和11阶多项式产生的数据进行拟合，生成图3.9所示的三种图形，从图形中可看出，多项式的阶数越高，拟合程度越高。

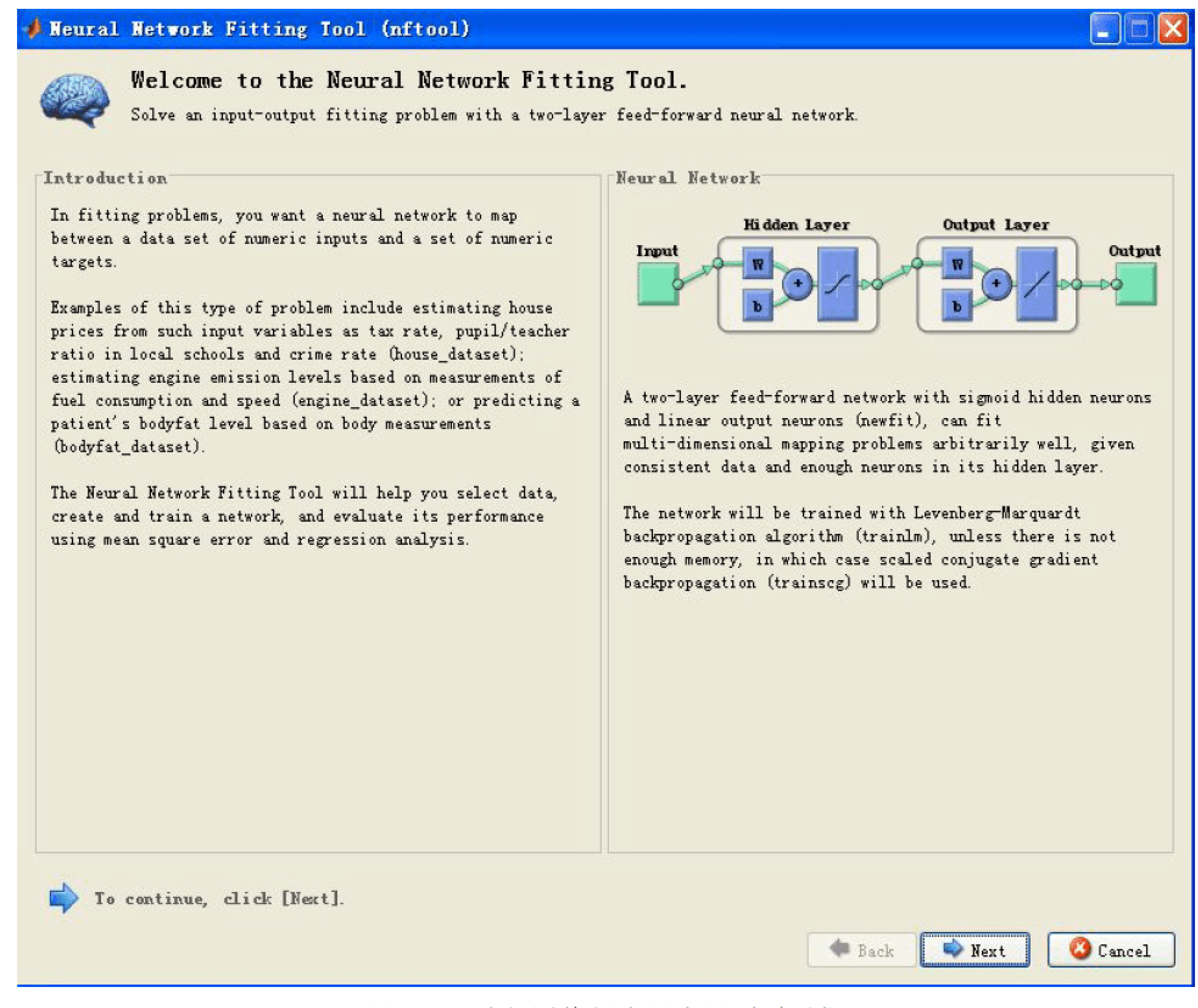
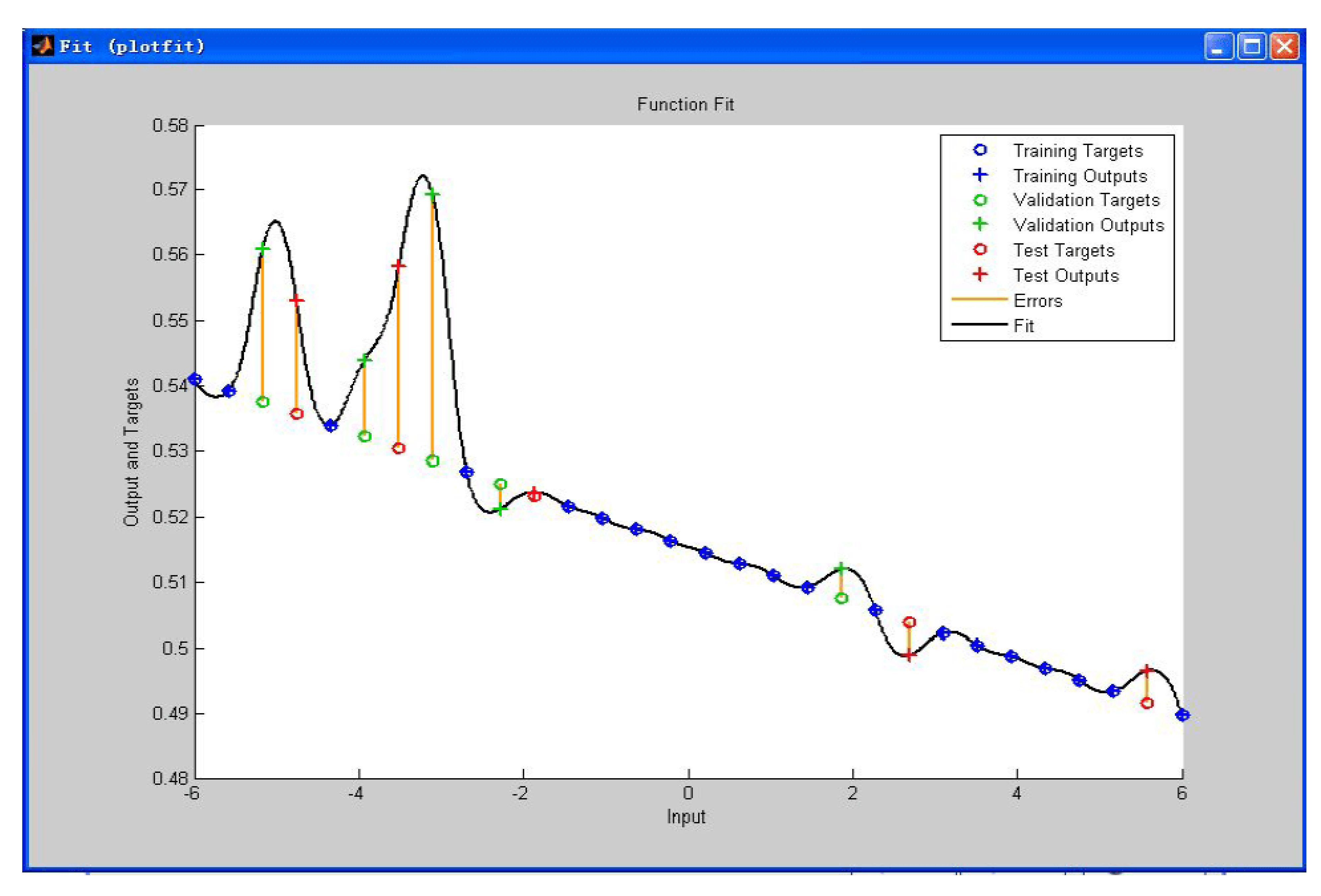
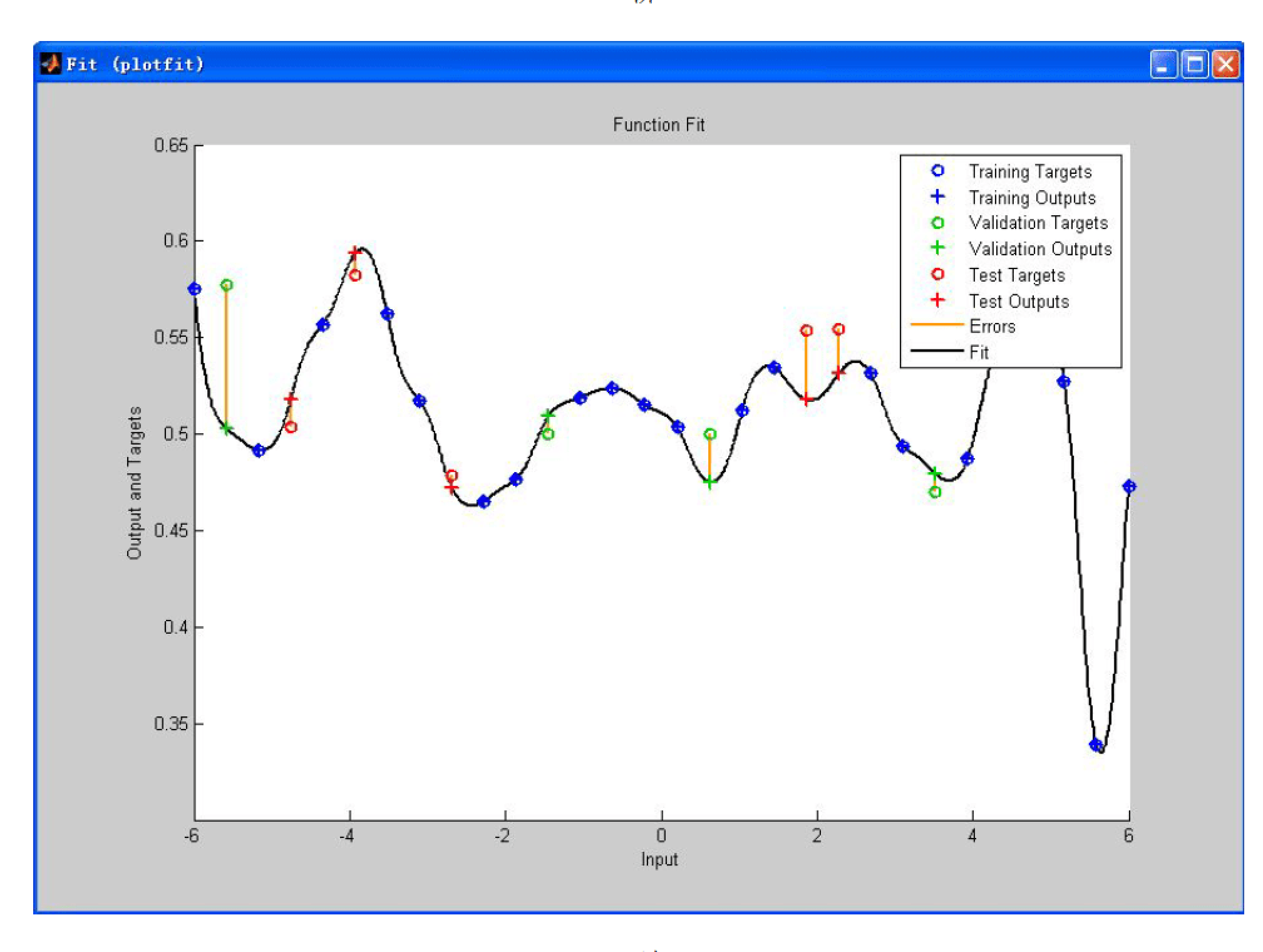


图 3.8　神经网络拟合用户图形对话框

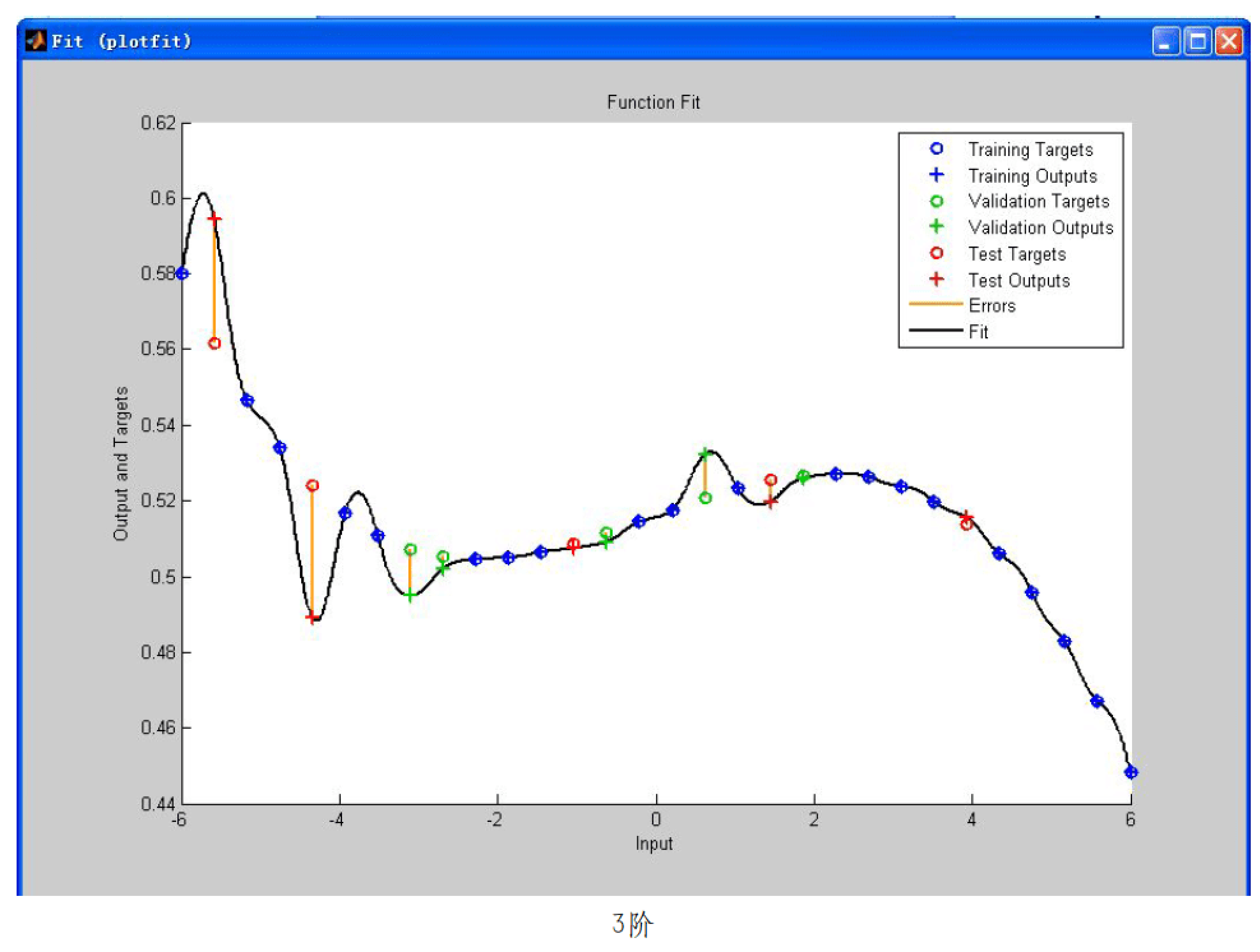


1 阶



11 阶

图 3.9　各阶多项式的拟合图形



3阶

续图 3.9

例 2.10　利用 BP 算法，研究以下各函数的逼近问题

$$f(x)=\frac{1}{x} \qquad 1\leqslant x\leqslant 100$$

解：

```
>> x=linspace(1,100,300);for i=1:length(x);y(i)=1/x(i);end;pr=minmax(x);
net=newff(pr,[10,1],{'logsig','logsig'});net.trainParam.show =
10;net.trainParam.lr = 0.05;
net.trainParam.goal = 1e-10;net.trainParam.epochs = 3000;net=train(net,x,y);
y1=sim(net,x);plot(x,y,'o',x,y1,'*-');hold on;ezplot('1/x',[1 100]);
```

得到图3.10的结果，可见逼近效果较好。

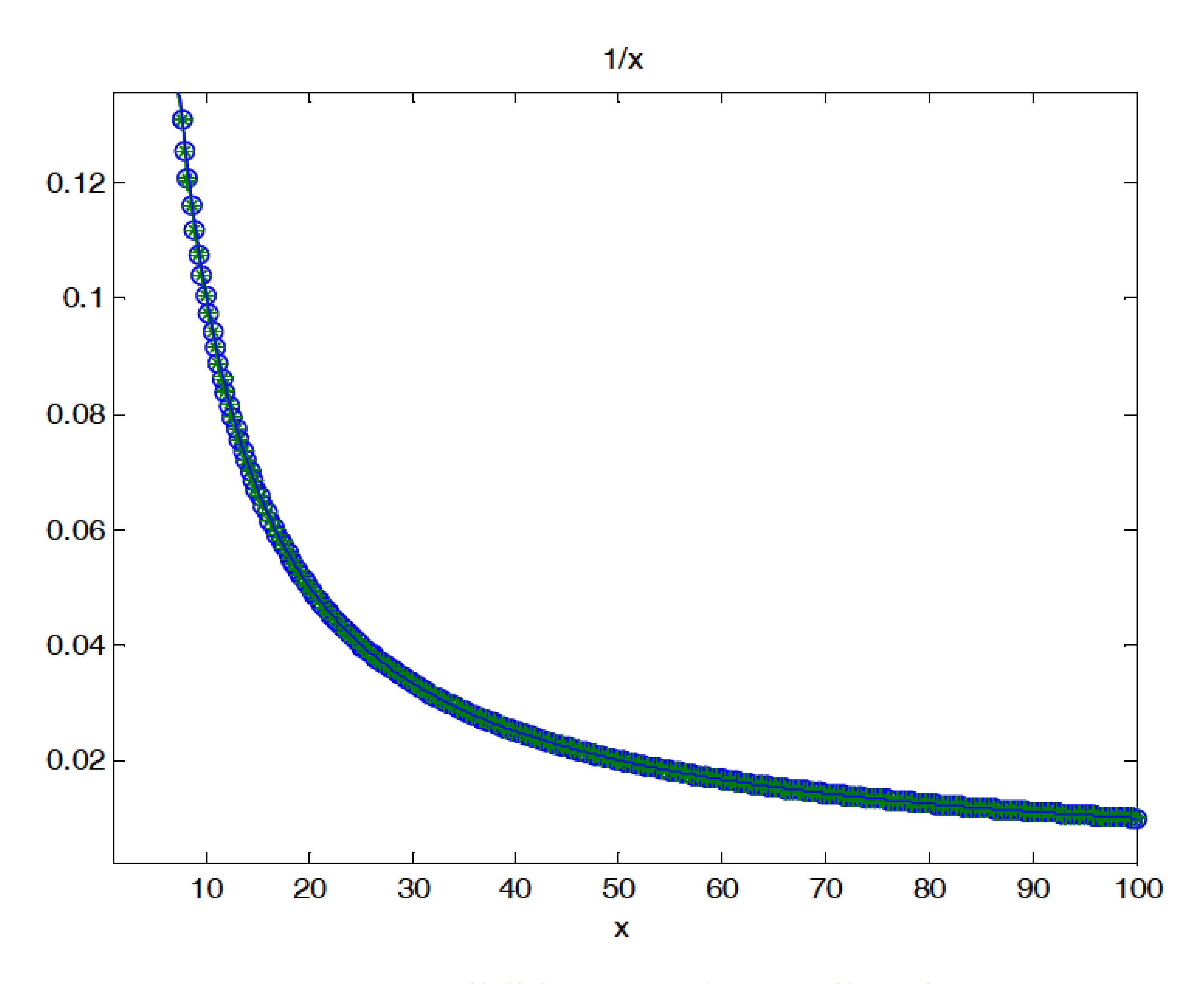


图 3.10　函数数据逼近图形及原函数图形

读书笔记

第 4 章

进化算法

4.1 概述

进化算法（Evolutionary Algorithm，EA）是一类模拟生物进化过程与机制求解问题的自组织、自适应人工智能技术。依照达尔文的自然选择和孟德尔的遗传变异理论，生物进化是通过繁殖、变异、竞争、选择来实现的。EA 算法就是建立在上述生物模型基础上的随机搜索技术。它采用编码来表示复杂的结构，并将每个编码称为一个个体（individual）。算法维持一定数量的编码集合，称为种群（population），并通过对种群中的个体进行一系列遗传操作（即交叉、变异和选择）来模拟进化过程，最终获得一些具有较高性能指标的编码。其中：交叉是模拟有性生殖过程中的染色体变换过程；变异是模拟自然界中生物遗传物质的变异；选择则是模拟自然界的优胜劣汰过程。

基因算法的典型实例有遗传算法、进化规划和进化策略等。

基因算法主要有以下一些名词。

- 个体（individual）：EA 所处理的基本对象、结构。
- 群体（population）：个体的集合称为种群体，该集合内个体的数量称为群体的大小。
- 串（bit string）：个体的表现形式，对应于生物界的染色体。在算法中其形式可以是二进制的，也可以是实数型。
- 基因（gene）：基因串中的元素，用于表示串中个体的特征。例如有一个串 $S_{二进制}=1011$，则其中的 1、0、1、1 这 4 个元素分别称为基因，它们的值称为等位基因（alletes）。一个个体的适应度函数值就是它的得分或评价。
- 基因位置（gene position）：一个基因在串中的位置称为基因位置，有时也简称基因位。基因位置由串的左向右计算，例如在串 $S_{二进制}=1101$ 中，0 的基因位置是 3。基因位置对应于遗传学中的地点（locus）。
- 基因特征值（gene feature）：在用串表示整数时，基因的特征值与二进制数的权一致。例如在串 S=1011 中，基因位置 3 中的 1，它的基因特征值为 2；基因位置 1 中的 1，它的基因特征值为 8。
- 串结构空间（bit string space）：在串中，基因任意组合所构成的串的集合，基因操作是在串结构空间中进行的。串结构空间对应于遗传学中的基因型(genotype)的集合。
- 参数空间（parameters space）：这是串空间在物理系统中的映射，它对应于遗传学中的表现型（Phenotype）的集合。
- 适应度及适应度函数（fitness）：适应度表示某一个体对于生存环境的适应程度，其值越大即对生存环境适应程度较高的物种将获得更多的繁殖机会；反之，其繁殖机会相对较少，甚至逐渐灭绝。适应度函数则是优化目标函数。
- 多样性或差异（diversity）：一个种群中各个个体间的平均距离。若平均距离大，则种群具有高的多样性；否则，其多样性低。多样性是进化算法必不可少的本质属性，它能使进化算法能搜索一个比较大的解的空间区域。
- 父代和子代：为了生成下一代，进化算法在当前种群中选择某些个体（称为父代），并且使用它们来生成下一代中的个体（称为子代）。典型情况下，算法更可能选择那些具有较佳适应度函数值的父代。
- 遗传算子：即进化算法中的算法规则，主要有选择算子、交叉算子和变异算子。

4.2　进化算法的基本原理

进化算法是借鉴生物界自然进化过程与机制而产生的一类随机搜索方法，它们是模拟由个体组成的群体的集体学习过程。其中每个个体表示给定问题搜索空间中的一点。进化算法从任一初始群体出发，通过选择、变异和交叉过程，使群体进化到搜索空间中越来越好的区域。选择过程使群体中适应性较好的个体比适应性差的个体有更多的生存机会；交叉过程使得子代继承父代的基因信息；而变异过程则是在群体中引入新的个体。

进化算法中主要涉及编码、适应度函数和遗传算子等基本要素。

4.2.1　编码

所谓编码，就是将问题的解空间转换成基因算法所能处理的搜索空间。编码是应用进化算法时要解决的首要问题，也是关键问题。它决定了个体的染色体中基因的排列次序，也决定了遗传空间到解空间的变换解码方法。编码的方法也影响到进化算子的计算方法，好的编码方法能够大大提高遗传算法的效率。

进化算法的工作对象是字符串，因此对字符串的编码有两点要求：一是字符串要反映所研究问题的性质；二是字符串的表达要便于计算机处理。常用的编码方法有以下几种。

1. 二进制编码

二进制编码是进化算法编码中最常用的方法。它是用固定长度的二进制符号{0，1}串来表示群体中的个体，个体中的每一位二进制字符称为基因。例如长度为 10 的二进制编码可以表示 0～1 023 之间的 1 024 个不同的数。如有一个待优化变量的区间[a，b]=[0，100]，则变量的取值范围可以被离散成 $(2^l)^p$ 个点，其中 l 为编码长度，p 为变量数目。从离散点 0 到离散点 100，依次对应于从 0000000000 到 0001100100。

二进制编码中符号串的长度与问题的求解精度有关。如果变量的变化范围为[a，b]，编码长度为 l，则编码精度为 $\dfrac{b-a}{2^l-1}$。

二进制编码、解码操作简单易行，杂交和变异等遗传操作便于实现，符合最小字符集编码原则，具有一定的全局搜索能力和并行处理能力。

2. 符号编码

符号编码是指个体染色体编码串中的基因值取自一个无数值意义而只有代码含义的符号集。这个符号集可以是一个字母表，如{A，B，C，D，…}；也可以是一个数字序列，如{1，2，3，4，…}；还可以是一个代码表，如{$A1$，$A2$，$A3$，$A4$，…}，等等。

符号编码符合有意义的积木块原则，便于在进化算法中利用所求问题的专业知识。

3. 浮点数编码

浮点数编码是指个体的每个基因用某一范围内的一个浮点数来表示。因为这种编码方法使用的是变量的真实值，所以也称为真值编码方法。

浮点数编码方法适合表示范围较大的数，适用于精度要求较高的进化算法，以便于在较大空间进行遗传搜索。

浮点数编码更接近于实际，并且可以根据实际问题来设计更有意义和与实际问题相关的交叉和变异算子。

4. 格雷编码

格雷编码是这样的一种编码，其连续的两个整数所对应的编码值之间仅有一个码位是不同的，其余的则完全相同。如 31 和 32 的格雷码为 010000 和 110000。格雷码与二进制编码之间有一定的对应关系。

设一个二进制编码为 $B=b_m b_{m-1}\cdots b_2 b_1$，则对应的格雷码为 $G=g_m g_{m-1}\cdots g_2 g_1$。由二进制向格雷码的转换公式为

$$g_i = b_{i+1} \oplus b_i,\quad i=m-1, m-2, \cdots, 1$$

由格雷码向二进制编码的转换公式为

$$b_i = b_{i+1} \oplus g_i,\quad i=m-1, m-2, \cdots, 1$$

其中：$\oplus$ 表示异与算子，即运算时两数相同时取 0、不同时取 1。如 $0\oplus 0=1\oplus 1=0,\ 0\oplus 1=1\oplus 0=1$。

使用格雷码对个体进行编码，编码串之间的一位差异，对应的参数值也只是微小的差异，这样与普通的二进制编码相比，格雷编码方法就相当于增强了进化算法的局部搜索能力，便于对连续函数进行局部空间搜索。

4.2.2 适应度函数

在用进化算法寻优之前，首先要根据实际问题确定适应度函数，即要明确目标。各个个体适应度值的大小决定了它们是继续繁衍还是消亡，以及能够繁衍的规模。它相当于自然界中各生物对环境的适应能力的大小，充分体现了自然界适者生存的自然选择规律。

与数学中的优化问题不同的是，适应度函数求取的是极大值，而不是极小值，并且适应度函数具有非负性。

对于整个进化算法影响最大的是编码和适应度函数的设计。好的适应度函数能够指导算法从非最优的个体进化到最优个体，并且能够用来解决一些遗传算法中的问题，如过早收敛与过慢结束。

过早收敛是指算法在没有得到全局最优解之前，就已稳定在某个局部解。其原因是因为某些个体的适应度值大大高于个体适应度的均值，在得到全局最优解之前，它们就有可能被大量复制而占群体的大多数，从而使算法过早收敛到局部最优解，失去了找到全局最优解的机会。解决的方法是压缩适应度的范围，防止过于适应的个体过早地在整个群体中占据统治地位。

过慢结束是指在迭代许多代后，整个种群已经大部分收敛，但是还没有得到稳定的全局最优解。其原因是因为整个种群的平均适应度值较高，而且最优个体的适应度值与全体适应度均值间的差异不大，使得种群进化的动力不足。解决的方法是扩大适应度函数值的范围，拉大最优个体适应度值与群体适应度均值的距离。

通常适应度函数是费用、盈利、方差等目标的表达式。在实际问题中，有时希望适应度越大越好，有时要求适应度越小越好。但在进化算法中，一般是按最大值处理，而且不允许适应度小于零。

对于有约束条件的极值，其适应度函数可用搜索空间限定法、可行解变换法和罚函数三种方法进行处理。

- 搜索空间限定法。对基因算法的搜索空间的大小加以限制，使搜索空间中表示一个个体的点与解空间中表示一个可行解的点有一一对应关系。进行搜索时，始终使算法在可行区域内。
- 可行解变换法。寻找出一种基因型和个体表现型之间的多对一的变换关系，使进化过程中产生的个体总能够通过这个变换转化成解空间中满足约束条件的一个可行解。
- 罚函数法。对解空间中无对应可行解的个体，计算其适应度时，给以一个罚函数，从而降低该个体适应度，使该个体被遗传到下一代群体中的机会减少，从而使该个体在群体中的更新换代中渐渐消失。

例如原来的极值问题为

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_i(x)=0, i=1,2,\cdots,n \\ & g_j(x)\geqslant 0, j=1,2,\cdots,m \end{aligned}$$

可转化为

$$\min \qquad f(x)+M\sum_{i=1}^{n}h_i^2(x)+M\sum_{j=1}^{m}[\min(0,g_j(x))]^2$$

式中：M 为惩罚系数。

4.2.3　遗传算子

遗传算子就是基因算法中进化的规则。基本基因算法的遗传算子主要有选择算子、交叉算子和变异算子。

1. 选择算子

选择算子就是用来确定如何从父代群体中按照某种方法，选择哪些个体作为子代的遗传算子。选择算子建立在对个体的适应度进行评价的基础上，其目的是为了避免基因的缺失，提高全局收敛性和计算效率。选择算子是 EA 的关键，体现了自然界中适者生存的思想。

常用选择算子的操作方法有以下几种。

（1）赌轮选择方法。

此方法的基本思想是个体被选择的概率与其适应度值大小成正比。为此，首先要构造与适应度函数成正比的概率函数 $p_s(i)$

$$p_s(i)=\frac{f(i)}{\sum\limits_{i=1}^{n}f(i)}$$

其中：$f(i)$为第 i 个个体适应度函数值，n 为种群规模。然后将每个个体按其概率函数 $p_s(i)$组成面积为 1 的一个赌轮。每转动一次赌轮，指针落入串 i 所占区域的概率即被选择复制的概率为 $p_s(i)$。当 $p_s(i)$较大时，串 i 被选中的几率大，但适应度值小的个体也有机会被选中，这样有利于保持群体的多样性。

（2）排序选择法。

排序选择法是指在计算每个个体的适应度值之后，根据适应度大小顺序对群体中的个体进行排序，然后按照事先设计好的概率表按序分配给个体，作为各自的选择概率。所有个体按适应度大小排序，选择概率和适应度无直接关系而仅与序号有关。

（3）最优保存策略。

此方法的基本思想是希望适应度最好的个体尽可能保留到下一代群体中。其步骤如下：

- 找出当前群体中适应度最高的个体和适应度最低的个体；
- 若当前群体中最佳个体的适应度比总的迄今为止的最好个体的适应度还要高，则以当前群体中的最佳个体作为新的迄今为止的最好个体；
- 用迄今为止的最好个体替换当前群体中最差个体。

该策略的实施可保证迄今为止得到的最优个体不会被交叉、变异等遗传算子破坏。

2. 交叉算子

交叉算子体现了自然界信息交换的思想，其作用是将原有群体的优良基因遗传给下一代，并生成包含更复杂结构的新个体。在交叉过程的开始，先产生随机数与交叉概率 p_c 比较，若随机数比 p_c 小，则进行交叉运算，否则不进行，直接返回父代。

交叉算子有一点交叉、二点交叉、多点交叉和一致交叉等。

（1）一点交叉。

首先在染色体中随机选择一个点作为交叉点，然后在第一个父辈的交叉点前串和第二个父辈交叉点后的串组合形成一个新的染色体，第二个父辈交叉点前的串和第一个父辈交叉点后的串形成另外一个新染色体。

例如下面两个串在第 5 位上进行交叉，生成的新染色体将替代它们的父辈而进入中间群体。

1010⊗xyxyyx ⎫
xyxy⊗xxxyxy ⎭ ⟶ 1010xxxyxy
xyxyxyxyyx

（2）二点交叉。

在父代中选择好两个染色体后，随机选择两个点作为交叉点。然后将这两个染色体中两个交叉点之间的字符串互换就可以得到两个子代的染色体。

例如下面两个串选择第 5 位和第 7 位为交叉点，然后交换两个交叉点间的串就形成两个新的染色体。

1010⊗xy⊗xyyx ⎫
xyxy⊗xx⊗xyxy ⎭ ⟶ 1010xxxyxy
xyxyxyxyyx

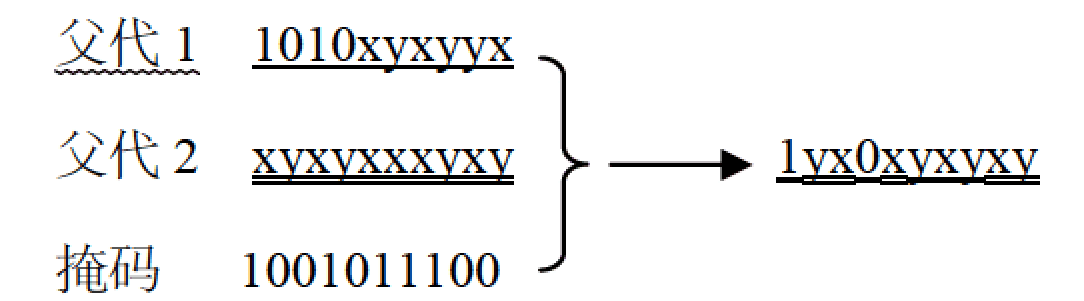
（3）多点交叉。

多点交叉与二点交叉相似。

（4）一致交叉。

在一致交叉中，子代染色体的每一位都是从父代相应位置随机复制而来，而其位置则由一个随机生成的交叉掩码决定。如果掩码的某一位是 1，则表示子代的这一位是从第一个父代中的相应位置复制，否则从第二个父代中相应位置复制。

例如下面父代按相应的掩码进行一致交叉：



3. 变异算子

变异算子是基因算法中保持物种多样性的一个重要途径，它模拟了生物进化过程中的偶然基因突变现象，其操作过程是对以变异概率随机指定的某个体编码位串的某一位或某几位基因位的基因值进行反运算，即由 1 变为 0，0 变为 1。

同自然界一样，每一位发生变异的概率是很小的，一般在 0.001～0.1 之间。如果过大，会破坏许多优良个体，也可能无法最优解。

基因算法的搜索能力主要是由选择和交叉赋予的。变异因子则保证了算法能搜索到问题解空间的每一点，从而使算法具有全局最优，进一步增强了 EA 的能力。

对产生的新一代群体进行重新评价选择、交叉和变异。如此循环往复，使群体中最优个体的适应度和平均适应度不断提高，直到最优个体的适应度达到某一限值或最优个体的适应度和群体的平均适应度不再提高，则迭代过程收敛，算法结束。

交叉概率、变异概率以及群体大小和遗传运算的终止进化代数的选择对基因算法的求解结果和效率有很大的影响。目前尚无合理选择选择参数的理论依据，在基因算法的实际应用中，往往需要经过多次试验后才能确定这些参数合理的取值大小或取值范围。

4.2.4　基因算法的特点

基因算法具有以下特点。

- 在生物系统中，进化被认为是一种成功的自适应方法，且具有很好的健壮性。
- 基因算法搜索的假设空间中，假设的各个部分相互作用，每一部分对总的假设适应度的影响难以建模。
- 基因算法易于并行性，且可降低由于使用超强计算机硬件所带来的昂贵费用。
- 基因算法采用一种随机化的搜索来寻找最大适应度的假设。这种搜索与其他很多学习方法的搜索完全不同。

4.3　基因算法的主要步骤

从假设的初始位串群体开始，基因算法按照以下步骤进行进化：

① 对问题进行编码；

② 定义适应度函数后，生成初始化群体；

③ 对于得到的群体进行选择复制，交叉，变异操作，生成下一代种群；

④ 判断算法是否满足停止准则。若不满足，则从步骤③起重复；

⑤ 算法结束，获得最优解。

整个操作过程可用图 4.1 来表示。

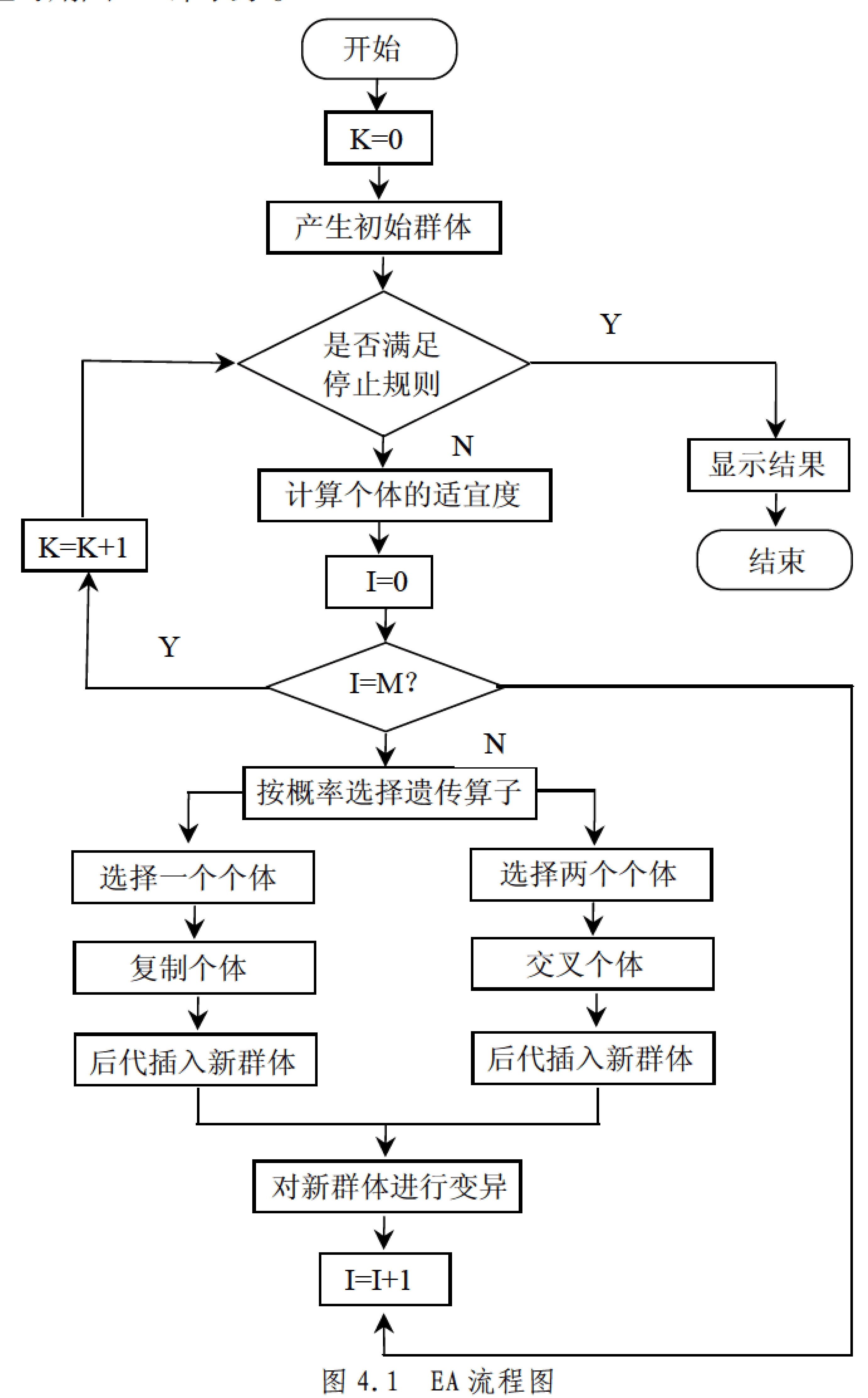


图 4.1 EA 流程图

4.4 基本遗传算法

基本遗传算法（Genetic Algorithm，GA）是最早出现的一种进化算法，它强调染色体的操作，即从一个初始种群出发，对种群中的每个个体（染色体）进行随机选择、交叉和变异操作，产生一群新的更适应环境的个体，使群体进化到搜索空间中越来越好的区域。这样代代的不断繁殖、进化，最后收敛到一群最适应环境的个体上，求得问题的最优解。

4.4.1 遗传算法的基本流程

遗传算法的基本流程如下。

① 编码：将问题解空间的可行解表示成遗传空间的基因型串结构数据，串结构数据的不同

组合构成了不同的可行解。

② 生成初始群体：随机产生 N 个初始串结构数据，每个串结构数据成为一个个体，N 个个体组成一个群体，遗传算法以该群体作为初始迭代点。

③ 适应度评估检测：根据实际标准计算个体的适应度，评判个体的优劣，即该个体所代表的可行解的优劣。

④ 进行选择操作：利用选择算子，选择当前群体中优良的（适应度高的）个体，使它们有机会被选中而进入下一次迭代；舍弃适应度低的个体，体现了进化论的“适者生存”的原则。

⑤ 进行交叉操作：对被选择的群体进行交叉算子操作，体现了信息交换的原则。

⑥ 进行变异操作：随机选择种群中的某个个体，以变异概率的大小改变个体某位基的值。变异为产生新个体提供了机会。

遗传算法的基本流程图如图 4.2 所示。

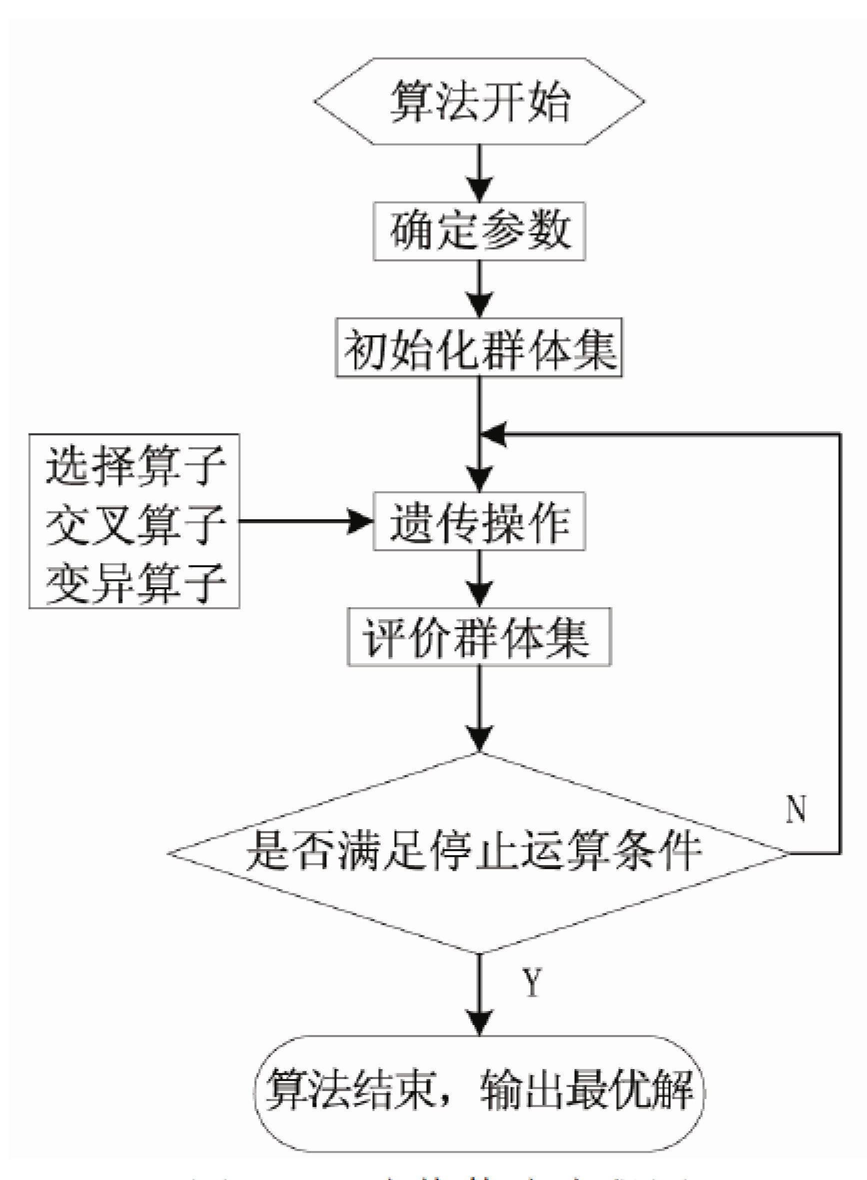


图 4.2　遗传算法流程图

4.4.2　控制参数选择

GA 中需要选择的参数主要有串长 l、群体大小 n、交叉概率 p_c 以及变异概率 p_m 等。这些参数对 GA 的性能影响较大。

1. 串长 l

串长的选择取决于特定问题解的精度。要求精度越高，串长越长，但需要更多的计算时间。为了提高运行效率，可采用变长度串的编码方式。

2. 群体大小 n

群体大小的选择与所求问题的非线性程度相关，非线性越大，n 越大。n 越大，则可以含有较多的模式，为遗传算法提供了足够的模式采样容量，改善遗传算法的搜索质量，防止成熟前收

敛，但也增加了计算量。一般建议取 n=20～200。

3. 交叉概率 p_c

交叉概率控制着交叉算子的使用频率。在每一代新群体中，需要对 $p_c \times n$ 个个体的染色体结构进行交叉操作。交叉概率越高，群体中新结构的引入就越快，同时，已是优良基因的丢失速率也相应提高了；而交叉概率太低则可能导致搜索阻滞。一般取 $p_c = 0.6 \sim 1.0$。

4. 变异概率 p_m

变异概率是群体保持多样性的保障。变异概率太小，可能使某些基因位过早地丢失信息而无法恢复，而太高则遗传算法将变成随机搜索。一般取 $p_m = 0.005 \sim 0.05$。

在简单遗传算法或标准遗传算法中，这些参数是不变的。但事实上这些参数的选择取决于问题的类型，并且需要随着遗传进程而自适应变化。只有这种有自组织性能的 GA 才能具有更高的鲁棒性、全局最优性和效率。

4.5 进化规划算法

作为进化计算的一个重要分支，进化规划算法具有进化计算的一般流程。在进化规划中，用高斯变异方法代替平均变异方法，以实现种群内个体的变异，保持种群中丰富的多样性。在选择操作上，进化规划算法采用父代与子代一同竞争的方式，采用锦标赛选择算子最终选择适应度较高的个体，其基本流程如图 4.3 所示。与其他进化算法相比，进化规划有其特点，它使用交叉、重组之类体现个体之间相互作用的算子，而变异算子是最重要的算子。

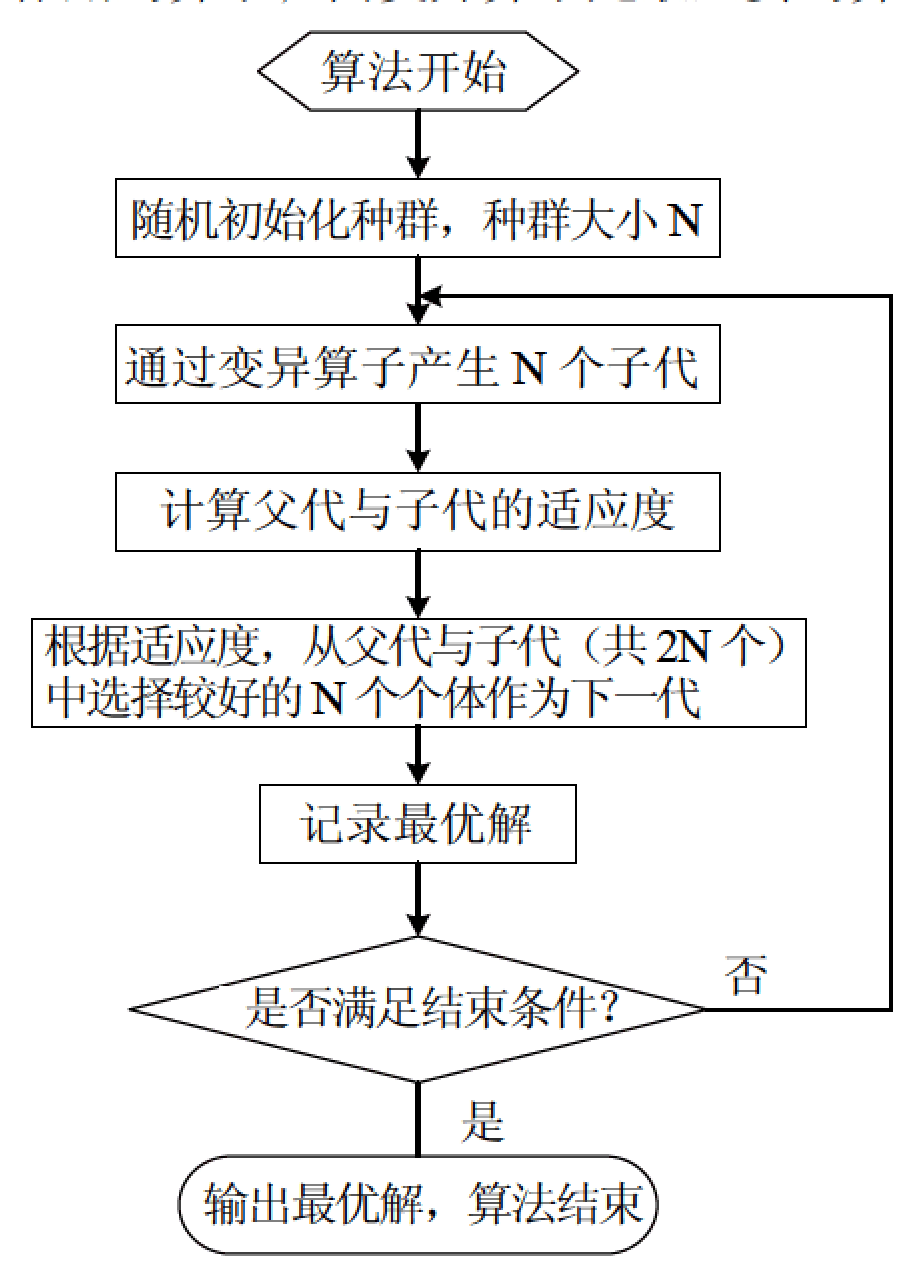


图 4.3　进化规划算法流程图

进化规划可应用于组合优化问题和复杂的非线性优化问题，它只要求所求问题是可计算的，使用范围比较广。

进化规划算法中的算子有变异算子、选择算子。

4.5.1　变异算子

在标准进化规划算法中，变异操作使用的是高斯变异算子。在变异过程中，计算每个个体适应度函数值的线性变换的平方根获得该个体变异的标准差 σ_i，将每个分量加上一个服从正态分布的随机数。

设 X 为染色体个体解的目标变量，有 L 个分量（即基因位），在 t+1 时有

$$X(t+1)=X(t)+N(0,\sigma)$$

$$\sigma(t+1)=\sqrt{\beta F(X(t))+\gamma}$$

$$x_i(t+1)=x_i(t)+N(0,\sigma(t+1))$$

式中：σ 为高斯变异的标准差；x_i 为 X 的第 i 个分量；$F(X(t))$ 为当前个体的适应度值（在这里，越是接近目标解的个体适应度值越小）；$N(0,\sigma)$ 是概率密度为的 $p(\sigma)=\dfrac{1}{\sqrt{2\pi}}\exp\left(-\dfrac{\sigma^2}{2}\right)$ 的高斯随机变量，系数 β_i 和 γ_i 是待定参数，一般将它们的值分别设为 1 和 0。

根据以上的计算方法，就可以得到变量 X 的变异结果。

4.5.2　选择算子

在进化规划算法中，选择操作是按照一种随机竞争的方式，根据适应度函数值从父代和子代的 2N 个个体中选择 N 个较好的个体组成下一代种群。选择的方法有依概率选择、锦标赛选择和精英选择三种。锦标赛选择方法是比较常用的方法，其基本原理如下。

① 将 N 个父代个体组成的种群和经过一次变异运算后得到的 N 个子代个体合并，组成一个共含有 2N 个个体的集合 I。

② 对每个个体 $x_i \in I$，从 I 中随机选择 q 个个体，并将 q 个个体的适应度函数值与 x_i 的适应度函数值相比较，计算出这 q（$q \geqslant 1$）个个体中适应度函数值比 x_i 的适应度差的个体的数目 w_i，并把 w_i 作为 x_i 的得分，$w_i \in$（$0,1,\cdots,q$）。

③ 在所有的 2N 个个体都经过这个比较后，按每个个体的得分 w_i 进行排序，选择 N 个具有最高得分的个体作为下一代种群。

通过这个过程，每代种群中相对较好的个体被赋予了较大的得分，从而能保留到下一代的群体中。

为了使锦标赛选择算子能发挥作用，需要适当地设定 q 值。q 值较大时，算子偏向确定性选择，当 q=2N 时，算子确定地从 2N 个个体中选择 N 个适应度较高的个体，容易造成早熟等弊端；相反，q 的取值较小时，算子偏向于随机性选择，使得适应度的控制能力下降，导致大量低适应度值的个体被选出，造成种群退化。因此，为了既能保持种群的先进性，又能避免确定性选择带来的早熟等弊病，需要根据具体问题，合理地选择 q 值。

4.6　进化策略计算

20 世纪 60 年代，德国柏林大学的 I.Rechenberg 和 H.P.Schwefel 等在进行风洞试验时，由于

设计中描述物体形状的参数难以用传统的方法进行优化，因而利用生物变异的思想来随机改变参数值，获得了较好的结果。随后，他们对这种方法进行了深入的研究和发展，形成了一种新的进化计算方法—进化策略。

在进化策略算法中，采用重组算子、高斯变异算子实现个体更新。1981 年，Schwefel 在早期研究的基础上，使用多个亲本和子代，后来分别构成（$\mu+\lambda$）－ES 和（μ,λ）－ES 两种进化策略算法。在（$\mu+\lambda$）－ES 中，由 μ 个父代通过重组和变异，生成 λ 个子代，并且父代与子代个体均参加生存竞争，选出最好的 μ 个个体作为下一代种群。在（μ,λ）－ES 中，由 μ 个父代生成 λ 个子代后，只有 λ（$\lambda>\mu$）个子代参加生存竞争，选择最好的 μ 个个体作为下一代种群，代替原来的 μ 个父代个体。

进化策略是专门为求解参数优化问题而设计的，而且在进化策略算法中引入了自适应机制。进化策略是一种自适应能力很好的优化算法，因此更多被应用于实数搜索空间。进化策略在确定了编码方案、适应度函数及遗传算法以后，算法将根据“适者生存，不适者淘汰”的策略，利用进化中获得的信息自行组织搜索，从而不断地向最佳方向逼近，隐含并行性和群体全局搜索性这两个显著特征，而且较强的鲁棒性，对于一些复杂的非线性系统求解具有独特的优越性能。

4.6.1 进化策略算法的基本流程

进化策略算法的流程如图 4.4 所示。

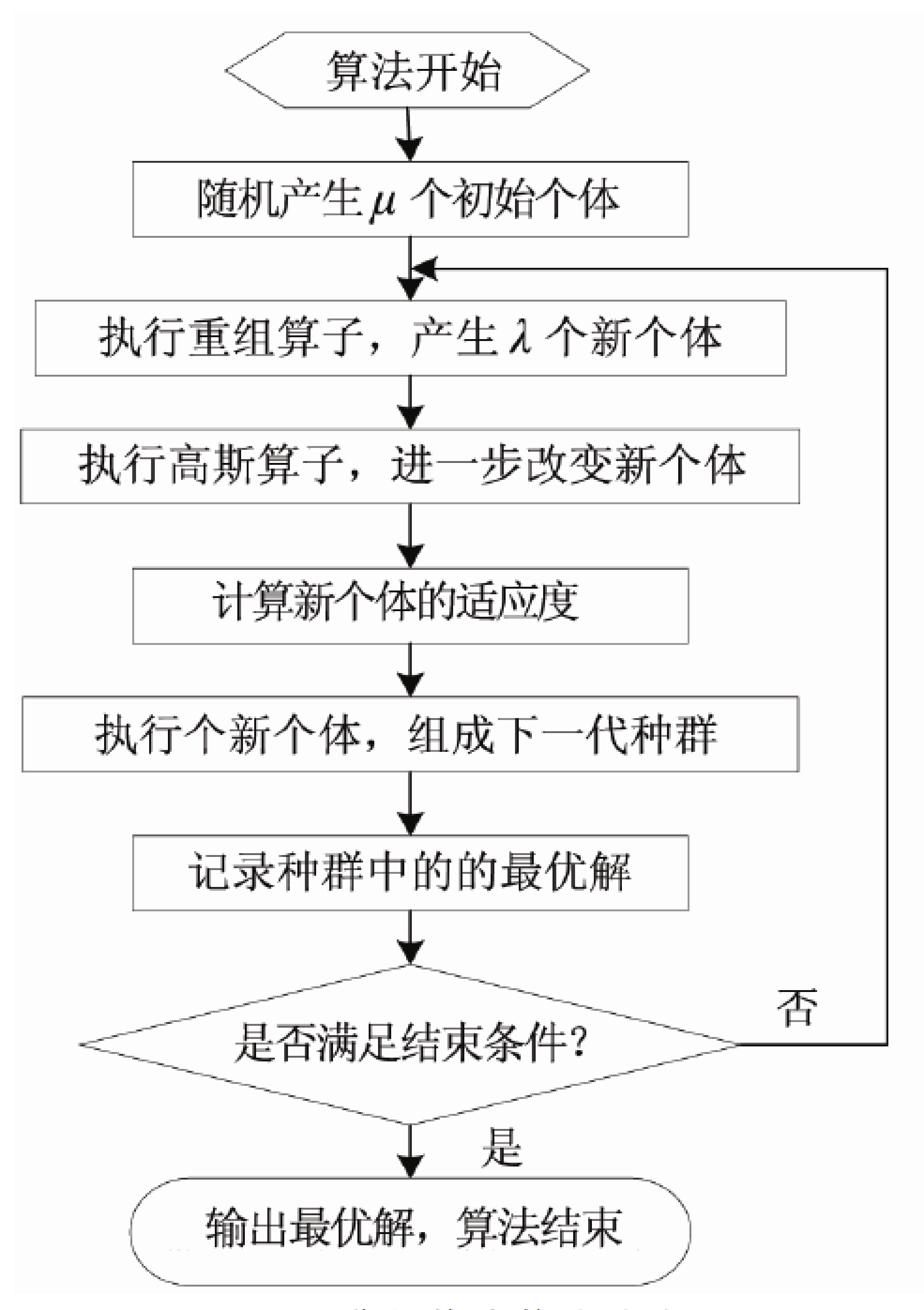


图 4.4 进化策略算法的流程图

4.6.2 算法的构成要素

1. 染色体构造

在进化策略算法中，常采用传统的十进制实数型表达问题，并且为了配合算法中高斯变异算

子的使用，染色体一般用以下二元表达方式

$$(X,\sigma) = ((x_1,x_2,\cdots,x_L),(\sigma_1,\sigma_2,\cdots,\sigma_L))$$

式中：X 为染色体个体的目标变量；σ 为高斯变异的标准差。每个 X 有 L 个分量，即染色体的 L 个基因位。每个 σ 有对应的 L 个分量，即染色体每个基因位的方差。

2. 进化策略的算子

（1）重组算子。

重组是将参与重组的父代染色体上的基因进行交换，形成下一代的染色体的过程。目前常见的有离散重组、中间重组、混杂重组等重组算子。

① 离散重组。

离散重组是随机选择两个父代个体来进行重组产生新的子代个体，子代上的基因随机从其中一个父代个体上复制。

两个父代：

$$(X^i,\sigma^i) = ((x_1^i,x_2^i,\cdots,x_L^i),(\sigma_1^i,\sigma_2^i,\cdots,\sigma_L^i))$$
$$(X^j,\sigma^j) = ((x_1^j,x_2^j,\cdots,x_L^j),(\sigma_1^j,\sigma_2^j,\cdots,\sigma_L^j))$$

然后将其分量进行随机交换，构成子代新个体的各个分量，从而得到以下的新个体

$$(X,\sigma) = ((x_1^{iorj},x_2^{iorj},\cdots,x_L^{iorj}),(\sigma_1^{iorj},\sigma_2^{iorj},\cdots,\sigma_L^{iorj}))$$

很明显，新个体只含有某一个父代个体的因子。

② 中间重组。

中间重组是通过对随机两个父代对应的基因进行求平均值，从而得到子代对应基因的方法，进行重组产生子代个体。

两个父代：

$$(X^i,\sigma^i) = ((x_1^i,x_2^i,\cdots,x_L^i),(\sigma_1^i,\sigma_2^i,\cdots,\sigma_L^i))$$
$$(X^j,\sigma^j) = ((x_1^j,x_2^j,\cdots,x_L^j),(\sigma_1^j,\sigma_2^j,\cdots,\sigma_L^j))$$

新个体：

$$(X,\sigma) = (((x_1^i+x_1^j)/2,(x_2^i+x_2^j)/2,\cdots,(x_L^i+x_L^j)/2),((\sigma_1^i+\sigma_1^j)/2,(\sigma_2^i+\sigma_2^j)/2,\cdots,(\sigma_L^i+\sigma_L^j)/2))$$

这时，新个体的各个分量兼容两个父代个体信息。

③ 混杂重组。

混杂重组方法的特点是在父代个体的选择上。混杂重组时先随机选择一个固定的父代个体，然后针对子代个体每个分量再从父代群体中随机选择第二个父代个体，也即第二个父代个体是经常变化的。

至于父代个体的组合方式既可以采用离散方式，也可以采用中值方式，甚至可以把中值重组中的 1/2 改为[0,1]之间的任一权值。

（2）变异算子。

变异算子的作用是在搜索空间中随机搜索，从而找到可能存在于搜索空间中的优良解。但若变异概率过大，则使搜索个体在搜索空间内大范围跃迁，使得算法的启发性和定向性作用不明显，

随机性增强，算法接近于完全的随机搜索；而若变异概率过小，则搜索个体仅在很小的领域范围内变动，发现新基因的可能性下降，优化效率很难提高。

进化策略的变异是在旧个体的基础上增加一个正态分布的随机数，从而产生新个体。

设 X 为染色体个体解的目标变量，有 L 个分量（即基因位）、σ 为高斯变异的标准差，在 t+1 时有

$$X(t+1)=X(t)+N(0,\sigma)$$

即

$$\sigma_i(t+1)=\sigma_i(t)\bullet\exp(N(0,\tau')+N_i(0,\tau))$$
$$x_i(t+1)=x_i(t)+N(0,\sigma_i(t+1))$$

式中：$(x_i(t),\sigma_i(t))$ 为父代个体第 i 个分量；$(x_i(t+1),\sigma_i(t+1))$ 为子代个体的第 i 个分量，$N(0,1)$ 是服从标准正态分布的随机数；$N_i(0,1)$ 是针对第 i 个分量生产一次符合标准正态分布的随机数；τ' 、τ 是全局系数和局部系数，通常都取 1。

（3）选择算子。

选择算子为进化规定了方向，只有具有高适应度的个体才有机会进行进化繁殖。在进化策略中，选择过程是确定性的。

在不同的进行策略中，选择机制也有所不同。

在（μ+λ）－ES 策略中，在原有 μ 个父代个体及新产生的 λ 个新子代个体中，再择优选择 μ 个个体作为下一代群体，即精英机制。在这个机制中，上一代的父代和子代都可以加入到下一代父代的选择中，$\mu>\lambda$ 和 $\mu=\lambda$ 都是可能的，对子代数量没有限制，这样就最大限度地保留了那些具有最佳适应度的个体，但是它可能会增加计算量，降低收敛速度。

在（μ，λ）－ES 策略中，因为选择机制依赖于出生过剩的基础上，因此要求 $\mu>\lambda$。在新产生的 λ 个新子代个体中择优选择 μ 个个体作为下一代父代群体。无论父代的适应度和子代相比是好是坏，在下一次迭代时都被遗弃。在这个机制中，只有最新产生的子代才能加入选择机制中，从 λ 中选择出最好的 μ 个个体，作为下一代的父代，而适应度较低的 $\lambda-\mu$ 个个体被放弃。

以上三种进化算法在本质上是相同的，但它们之间又存在区别。其中，进化规划与遗传算法的区别主要体现在以下三个方面：

① 在待求问题的表示方面，进化规划因为其变异操作不依赖于线性编码，所以往往可以根据待求问题的具体情况而采取一种较为灵活的组织方式；而遗传算法则通常要把问题的解编码成一串表达符号，即基因串的形式。前者更类似于人工神经网络对问题的表达方式。

② 在后代个体的产生方面，进化规划侧重于群体中个体行为的变化。与遗传算法所不同的是，它没有利用个体之间的信息变换，所以也就省去了交叉算子而只保留了变异操作。因此在不考虑效率的前提下，进化规划算法在应用方面更易于掌握，便于实现。

③ 在竞争与选择方面，进化规划允许父代与子代一起参与竞争，正因为如此，进化规划可以保证以概率 1 收敛的全局最优解；而若不强制父代最佳解的典型遗传算法，是不收敛的。

进化规划与进化策略的主要区别体现在以下两个方法。

① 在编码结构方面，进化规划是将种群变化类比为编码结构，而进化策略则是把个体类比于编码结构。所以，前者不需要再通过选择操作来产生新候选解，而后者还要进行这一操作。

② 在竞争与选择方面，进化规划要通过适当的选择机制，从父代和当前子代中选取优胜者组成下一代群体；而进化策略则是通过一种确定性选择，按适应值大小直接将当前优秀个体和父代中的最佳个体保留到下一代。

4.7　量子遗传算法

分析 EA 算法可以发现，它没有利用进化中未成熟优良子群体所提供的信息，因而限制了进化速度。事实证明，在进化中引入好的引导机制可以增强算法的智能性，提高搜索效率，解决 EA 中的早熟和收敛速度问题。现有 EA 的许多改进工作也正是致力于这一方面。

量子遗传算法（Quantum Genetic Algorithm，QGA）是量子进化理论与 EA 算法结合的产物。量子计算具有天然的并行性，极大地加快了对海量信息处理的速度，使得大规模复杂问题能够在有限的指定时间内完成。利用量子计算的这一思想，将量子算法与经典算法相结合，通过对经典表示方法进行相应的调整，使得其具有量子理论的优点，从而成为有效的算法。

量子遗传算法使用量子比特编码染色体。这种概率幅表示可以使一个量子染色体同时表征多个状态的信息，带来丰富的种群，而且当前最优个体的信息能够很容易地用来引导变异，使得种群以大概率向着优良模式进化，加快收敛。

4.7.1　基本概念

1. 量子比特

用量子比特来存储和处理信息，称为量子信息。区别量子信息与经典信息最大的不同在于：经典信息，比特只能处在一个状态，非 0 即 1；而在量子信息中，量子比特可以同时处在$|0\rangle$和$|1\rangle$两个状态，量子信息的存储单元称为量子比特（qubit）。一个量子比特的状态是一个二维复数空间的矢量，它的两个极化状态$|0\rangle$和$|1\rangle$对应于经典状态的 0 和 1。

量子比特不仅可以表示 0 和 1 两种状态，也可以同时表示两个量子的叠加态，即“0”态和“1”态的任意中间态。一般情况下，用 n 个量子位就可以同时表示 2^n 个状态，其叠加态可以描述为

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

式中：(α,β) 是一对复数，表示相应比特状态的概率幅，且满足归一化条件，即$|\alpha|^2+|\beta|^2=1$，$|0\rangle$和$|1\rangle$分别表示两个不同的比特态，且$|\alpha|^2$表示$|0\rangle$的概率，$|\beta|^2$表示$|1\rangle$的概率。利用不同的量子叠加态记录不同的信息，量子比特在同一位置可拥有不同的信息。

2. 量子染色体

在 QGA 中，使用基于量子比特编码方式，即用一对复数定义一个量子比特位。一个具有 m 个量子比特位的系统可以描述为：$\begin{bmatrix}\alpha_1|\alpha_2|\cdots\alpha_m\\ \beta_1|\beta_2|\cdots\beta_m\end{bmatrix}$，其中$|\alpha_i|^2+|\beta_i|^2=1$，$i=1,2,\cdots,m$。因此，染色体种群中第 t 代的个体 X_j^t 可表示为 $X_j^t=\begin{bmatrix}\alpha_1^t|\alpha_2^t|\cdots\alpha_m^t\\ \beta_1^t|\beta_2^t|\cdots\beta_m^t\end{bmatrix}$ $(j=1,2,\cdots,m)$，其中 N 为种群大小，t 为进化代数。

4.7.2 量子遗传算法流程

量子遗传算法是在传统的遗传算法中引入量子计算的概念和机制后形成的新算法。与传统的遗传算法一样，量子遗传算法中也包括个体种群的构造、适应度值的计算、个体的改变，以及种群的更新。而与传统遗传算法不同的是，量子遗传算法中的个体是包含多个量子位的量子染色体，具有叠加性、纠缠性等特性，一个量子染色体可呈现多个不同状态的叠加。通过不断的迭代，每个量子位的叠加态将坍塌到一个确定的态，从而达到稳定，趋于收敛。量子遗传算法就是通过这样的一个方式，不断地进行探索、进化，最后达到寻优的目的。

量子遗传算法的流程如图 4.5 所示，可分为以下各步骤。

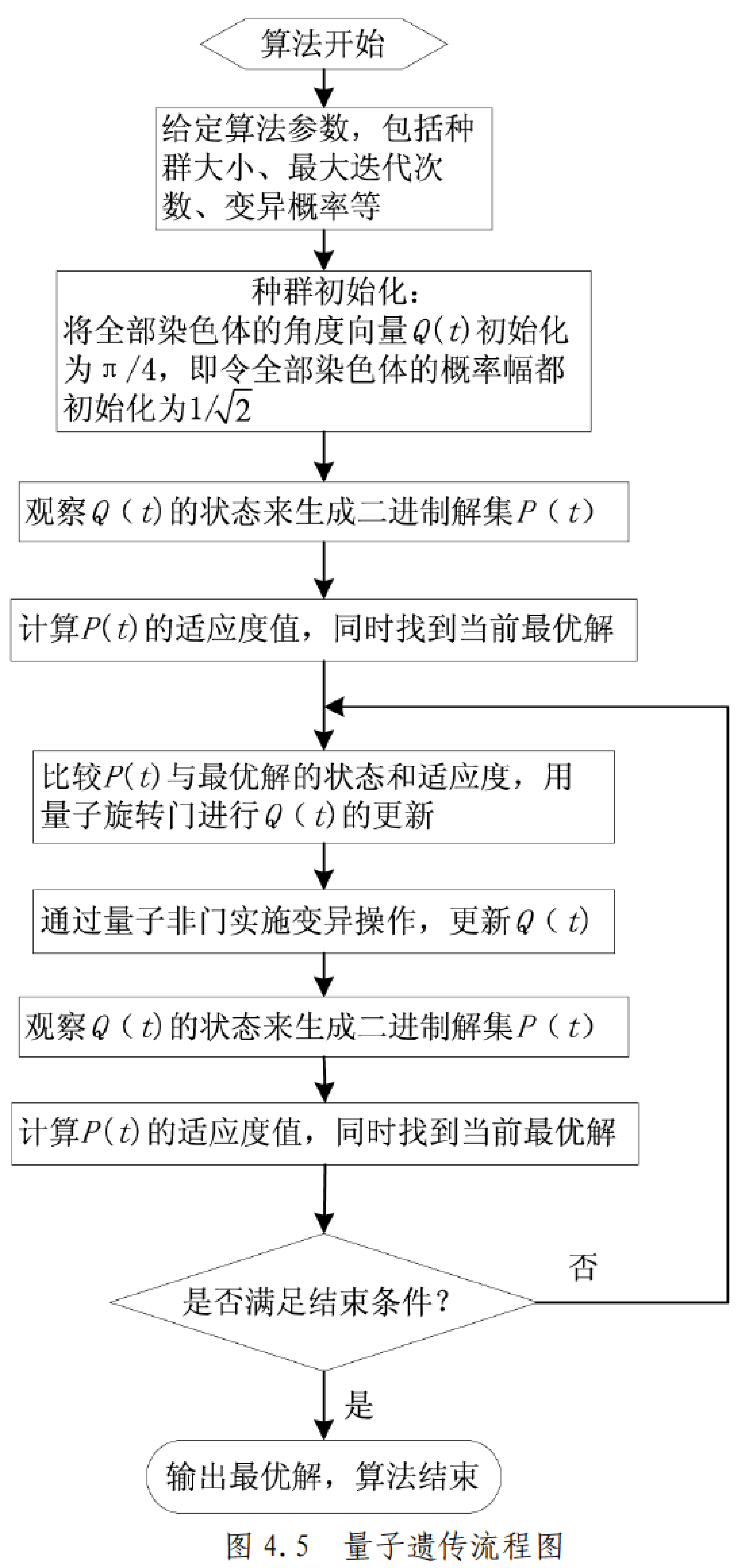


图 4.5　量子遗传流程图

① 给定算法参数，包括种群大小、最大迭代次数，交叉概率，变异概率。

② 种群初始化。

初始化 N 条染色体 $P(t)=(X_1^t,X_2^t,\cdots,X_N^t)$，将每条染色体 X_i^t 的每一个基因用二进制表示，每一个二进制位对应一个量子位。设每个染色体有 m 个量子位，$X_i^t=(x_{i1}^t,x_{i2}^t,\cdots,x_{im}^t)$ $(i=1,2,\cdots,N)$ 为一个长度为 m 的二进制串，有 m 个观察角度 $Q_i^t=(\varphi_{i1}^t,\varphi_{i2}^t,\cdots,\varphi_{im}^t)$，其值决定量子位的观测概率 $|\alpha_i^t|^2$ 或 $|\beta_i^t|^2$ $(i=1,2,\cdots,m)$，$\begin{pmatrix}\alpha_i\\\beta_i\end{pmatrix}=\begin{pmatrix}\cos(\varphi)\\\sin(\varphi)\end{pmatrix}$，通过观察角度 $Q(t)$ 的状态来生成二进制解集 $P(t)$。初始化使所有量子染色体的每个量子位的观察角度 $\varphi_{ij}^0=\dfrac{\pi}{4}$，其中 i=1,2,…,N；j=1,2,…,m；概率幅都初始化为 $\dfrac{1}{\sqrt{2}}$，它表示在 $t=0$ 代，每条染色体以相同的概率 $\dfrac{1}{\sqrt{2^m}}$ 处于所有可能状态的线性叠加态之中，即 $|\psi_{qj}^0\rangle=\sum\limits_{k=1}^{2m}\dfrac{1}{\sqrt{2^m}}|s_k\rangle$，其中 s_k 是由二进制串 $(x_1,x_2,\cdots,x_m)$ 描述的第 k 个状态。

③ 计算 $P(t)$中每个解的适应度，存储最优解。

④ 开始进入迭代。

⑤ 量子旋转门。量子旋转门操作是以当前最优解为引导的旋转角度作为量子染色体变异的表现，通过观测最优个体和当前个体相应量子位所处状态，以及比较它们的适应度值，来确定其旋转角度的变化方向和大小。量子门可根据实际问题具体设计，令 $U(\Delta\theta)=\begin{bmatrix}\cos(\Delta\theta) & -\sin(\Delta\theta)\\ \sin(\Delta\theta) & \cos(\Delta\theta)\end{bmatrix}$ 表示量子旋转门，设 φ 为原量子位的幅角，旋转后的角度调整操作为

$$\begin{pmatrix}\alpha_i'\\\beta_i'\end{pmatrix}=\begin{pmatrix}\cos(\Delta\theta) & -\sin(\Delta\theta)\\ \sin(\Delta\theta) & \cos(\Delta\theta)\end{pmatrix}\begin{pmatrix}\alpha_i\\\beta_i\end{pmatrix}=\begin{pmatrix}\cos(\varphi+\Delta\theta)\\\sin(\varphi+\Delta\theta)\end{pmatrix}$$

式中：$\begin{pmatrix}\alpha_i\\\beta_i\end{pmatrix}=\begin{pmatrix}\cos(\varphi)\\\sin(\varphi)\end{pmatrix}$ 为染色体中第 i 个量子位，且 $|\alpha_i|^2+|\beta_i|^2=1$，$\Delta\theta$ 为旋转角度。

⑥ 通过量子非门进行变异操作，更新 $P(t)$。为避免陷入早熟和局部极值，在此基础上进一步采用量子非门实现染色体变异操作，这样能够保持种群多样性和避免选择压力。

⑦ 通过观察角度 $Q(t)$ 的状态来生成二进制解集 $P(t)$，即对于每一个比特位，随机产生一个[0,1]之间的随机数 r。比较 r 与 $|\alpha_i^t|^2$ 的大小，如果 $r<|\alpha_i^t|^2$，则令该比特位值为 1；否则令其为 0。

⑧ 计算 $P(t)$的适应度值，最后选择 $P(t)$中的当前最优解，若该最优解于优于目前存储的最优解，则用该最优解替换存储的最优解，更新全局最优解。

⑨ 判断是否达到最大迭代次数，如果是，则跳出循环，输出最优解；否则，则转到步骤⑤，继续进行。

4.7.3　量子算法中的控制参数

1. 量子染色体

与传统进化算法不同，量子遗传算法不直接包含问题，而是引入量子计算中的量子位，采用

基于量子位的编码方式来构造量子染色体，以概率幅的形式来表示某种状态的信息。

一个量子位可由其概率幅定义为$\begin{bmatrix}\alpha\\\beta\end{bmatrix}$，同理 m 个量子位可定义为$\begin{bmatrix}\alpha_1|\alpha_2|\cdots\alpha_m\\\beta_1|\beta_2|\cdots\beta_m\end{bmatrix}$，其中 $|\alpha_i|^2+|\beta_i|^2=1$，$i=1,2,\cdots,m$。因此，染色体种群中第 t 代的个体 X_j^t 可表示为 $X_j^t=\begin{bmatrix}\alpha_1^t|\alpha_2^t|\cdots\alpha_m^t\\\beta_1^t|\beta_2^t|\cdots\beta_m^t\end{bmatrix}(j=1,2,\cdots,m)$，其中 N 为种群大小，t 为进化代数。

量子比特具有叠加性，因此通过量子位的概率幅产生新个体使得每一个比特位上的状态不再是固定的信息，一个染色体不再仅对应于一个确定的状态，而变成了一种携带着不同叠加态的信息。由于这种性质，使得基于量子染色体编码的进化算法，比传统遗传算法具有更好的种群多样性。经过多次迭代后，某一个量子比特上的概率幅$|\alpha|^2$或$|\beta|^2$趋近于 0 或 1 时，这种不确定性产生的多样性将逐渐消失，最终坍塌到一个确定状态，从而使算法最终收敛，这就表明量子染色体同时具有探索和开发两种能力。

2. 量子旋转门

在量子计算中，各个量子状态之间的转移变换主要是通过量子门实现的。而量子门对量子比特的概率幅角度进行旋转，同样可以实现量子状态的改变。因此，在量子遗传算法中，使用量子旋转门来实现量子染色体的变异操作。同时，由于在角度旋转时考虑了最优个体的信息，因此，在最优个体信息的指导下，可以使种群更好地趋向最优解，从而加快了算法收敛。在 0,1 编码的问题中，令$U(\Delta\theta)=\begin{bmatrix}\cos(\Delta\theta) & -\sin(\Delta\theta)\\\sin(\Delta\theta) & \cos(\Delta\theta)\end{bmatrix}$表示量子旋转门，旋转角度变异的角度 θ 可由表 4.1 得到。

表 4.1　变异角 θ（二值编码）

旋转角度				旋转角度符号 $s(\alpha_i\beta_i)$			
x_i	x_i^{best}	$f(X)\geqslant f(X^{best})$	$\Delta\theta_i$	$\alpha_i\beta_i>0$	$\alpha_i\beta_i<0$	$\alpha_i=0$	$\beta_i=0$
0	0	假	0	0	0	0	0
0	0	真	0	0	0	0	0
0	1	假	0	0	0	0	0
0	1	真	0.05π	−1	+1	±1	0
1	0	假	0.05π	−1	+1	±1	0
1	0	真	0.05π	+1	−1	0	±1
1	1	假	0.05π	+1	−1	0	±1
1	1	真	0.05π	+1	−1	0	±1

表中 x_i 为当前量子染色体的第 i 位，x_i^{best} 为当前最优染色体的第 i 位，均为观察值；$f(X)$ 为适应度函数，$\Delta\theta_i$ 为旋转角度的大小，控制算法收敛的速度，取值太小将造成收敛速度过慢，但太大可能会使结果发散，或“早熟”收敛到局部最优解。$\Delta\theta_i$ 取值可固定也可自适应调整大小；α_i，β_i 为当前染色体第 i 位量子位的概率幅；$s(\alpha_i\beta_i)$为旋转角度的方向，保证算法的收敛。

3. 量子非门操作

采用量子非门实现染色体的变异。首先从种群中随机选择出需要实施变异操作的量子染色

体，并在这些量子染色体的若干量子比特上实施变异操作。假设$\begin{bmatrix}\alpha_i\\ \beta_i\end{bmatrix}$为该染色体的第 i 个量子位，使用量子非门实施变异操作的过程可描述为

$$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\alpha_i\\ \beta_i\end{bmatrix}=\begin{bmatrix}\beta_i\\ \alpha_i\end{bmatrix}$$

由上式可以看出，量子非门实施的变异操作，实质上是量子位的两个概率幅互换。由于更改了量子比特态叠加的状态，使得原来倾向于坍塌到状态“1”变为倾向于坍塌到状态“0”，或者相反，因此起到了变异的作用。显然，该变异操作对染色体的所有叠加态具有相同的作用。

从另一角度看，这种变异同样是对量子位幅角的一种旋转：如假设某一量子位幅角为 q，则变异后的幅角变为$(\pi/2)-q$，即幅角正向旋转了 $\pi/2$。这种旋转不与当前染色体比较，一律正向旋转，有助于增加种群的多样性，降低“早熟”收敛的概率。

4.8　人工免疫算法

20 世纪 80 年代中期，美国 Michigan 大学的 Holland 教授提出的遗传算法，虽然具有使用方便、鲁棒性强、便于并行处理等特点，但在对算法的实施过程中不难发现两个主要遗传算子都是在一定发生概率的条件下，随机地、没有指导地迭代搜索，因此它们在为群体中的个体提供进化机会的同时，也不可避免地产生了退化的可能，在某些情况下，这种退化现象还相当明显。另外，每一个待求的实际问题都会有自身一些基本的、明显的特征信息或知识，然而，遗传算法的交叉和变异算子却相对固定，在求解问题时，可变的灵活程度较小，这无疑对算法的通用性是有益的，但却忽视了问题的特征信息对求解问题时的辅助作用，特别是在求解一些复杂问题时，这种忽视所带来的损失往往是比较明显的。实践也表明，仅仅使用遗传算法或者以其为代表的进化算法，在模仿人类智能处理事物的能力方面还远远不足，必须更加深层次地挖掘与利用人类的智能资源。所以，研究者力图将生命科学中的免疫概念引入到工程实践领域，借助其中的有关知识与理论并将其与已有的一些智能算法有机地结合起来，以建立新的进化理论与算法，来提高算法的整体性能。基于这个思想，将免疫概念及其理论应用于遗传算法，在保留原算法优良特性的前提下，力图有选择、有目的地利用待求问题中的一些特征信息或知识来抑制其优化过程出现的退化现象，这种算法称为免疫算法（Immune Algorithm，IA）。

4.8.1　人工免疫算法的生物学基础

1. 生物免疫系统

生物免疫系统是由具有免疫功能的器官、组织、细胞、免疫效应分子和有关的基因等组成。它是生物在不断的进化过程中，通过识别“自己”和“非己”，排除抗原性“异物”，保护自身免受致病细菌、病毒或病原性异物的侵袭，维持机体环境平衡，维护生命系统正常运作。生物免疫系统是机体的保护性生理反应，也是机体适应环境的体现，具有对环境不断学习、后天积累的功能，它的结构及其行为特性极为复杂，关于其内在规律的认识，人们仍在进行不懈的努力。

为了便于了解免疫系统的基本原理，促进基本免疫机理的算法和模型用于解决实际工程问题，有必要先简单介绍一些基本概念和技术术语。

（1）免疫淋巴组织。

免疫淋巴组织按照作用不同分为中枢淋巴组织和周围淋巴组织。前者包括胸腺、腔上囊、人类和哺乳类的相应组织是骨髓和肠道淋巴组织；后者包括脾脏、淋巴结和全身各处的弥散淋巴组织。

（2）免疫活性细胞。

免疫活性细胞是能受抗原刺激，并能引起特异性免疫反应的细胞。按发育成熟的部位及功能不同，免疫活性细胞分成 T 细胞和 B 细胞两种。

① T 细胞。

T 细胞又称胸腺依赖性淋巴细胞，由胸腺内的淋巴干细胞在胸腺素的影响下增殖分化而成，它主要分布在淋巴结的深皮质区和脾脏中央动脉的胸腺依赖区。T 细胞受抗原刺激时首先转化成淋巴细胞，然后分化成免疫效应细胞，参与免疫反应，其功能包括调节其他细胞的活动以及直接袭击宿主感染细胞。

② B 细胞。

B 细胞又称免疫活性细胞，由腔上囊组织中的淋巴干细胞分化而成，来源于骨髓淋巴样前体细胞，主要分布在淋巴结、血液、脾、扁桃体等组织和器官中。B 细胞受抗原刺激后，首先转化成浆母细胞，然后分化成浆细胞，分泌抗体，执行细胞免疫反应。

（3）抗原与抗体。

抗原一般是指诱导免疫系统产生免疫应答的物质，包括各种病原性异物以及发生了突变的自身细胞（如癌细胞）等。抗原具有刺激机体产生抗体的能力，也具有与其所诱生的抗体相结合的能力。

抗体又称免疫球蛋白，是指能与抗原进行特异性结合的免疫细胞，其主要功能是识别、消除机体内各种病原性异物。抗体可分为分泌型和膜型，前者主要存在于血液及组织液中，发挥各种免疫功能；后者构成 B 细胞表面的抗原受体。各种抗原分子都有其特异结构 Idiotype－抗原化学基，又称 Epitope－表位，而每个抗体分子 V 区也存在类似机构受体，或称 Paratope－对位。抗体根据其受体与抗原化学基的分子排列相互匹配情况识别抗原。当两种分子排列的匹配程度较高时，两者亲和度（Affinity）较大，亲和度大的抗体与抗原之间会产生生物化学反应，通过相互结合形成绑定（Banding）结构，并促使抗原逐步凋亡。

（4）亲和力。

免疫细胞表面的抗体和抗原化学基都是复杂的含有电荷的三维结构，抗体和抗原的结构与电荷之间互补就有可能结合，结合的强度即为亲和力。

（5）亲和力成熟。

数次活化后的子代细胞仍保持原代 B 细胞的特异性，但中间可能会发生重链的类转换或点突变，这两种变化都不影响 B 细胞对抗原识别的特异性，但点突变影响其产生抗体对抗原的亲和力。高亲和性突变的细胞有生长增殖的优先权，而低亲和性突变的细胞则选择性死亡，这种现象被称为亲和力成熟，它有利于保持在后继应答中产生高亲和性的抗体。

（6）变异。

在生物免疫系统中，B 细胞与抗原之间结合后被激活，然后产生高频变异。这种克隆扩增期

间产生的变异形式，使免疫系统能适应不断变化的外来入侵。

（7）免疫应答。

免疫应答是指抗原进入机体后，免疫细胞对抗原分子的识别、活化、分化和效应等过程；它是免疫系统各部分生理的综合体现，包括了抗原提呈、淋巴细胞活化、特异识别、免疫分子形成、免疫效应以及形成免疫记忆等一系列的过程。

（8）免疫耐受。

免疫耐受是指免疫活性细胞接触抗原物质时所表现的一种特异性的无应答状态。免疫耐受现象是指由于部分细胞的功能缺失或死亡而导致的机体对该抗原反应功能丧失或无应答的现象。

4.8.2　生物免疫基本原理

抗原入侵机体后会刺激免疫系统发生一系列复杂的连锁反应，这个过程即为免疫应答或称免疫反应。

免疫应答有两种类型：一种是遇到病原体后首先并迅速起防卫作用的固有性免疫应答；另一种是适应性免疫应答。前者在感染早期执行防卫功能；后者是继固有性免疫应答之后发挥效应的，以最终消除病原体，促进疾病治愈及防止再感染起主导作用。

适应性免疫应答又分为初次应答和二次应答。

抗原初次进入机体后，免疫系统就产生应答（初次应答），通过刺激有限的特异性克隆扩增，迅速产生抗体，以达到足够的亲和力阈值，消除抗原，并对其保持记忆，以便下次遭到同样的抗原时更加快速地做出应答。初次应答比较慢，使得免疫系统有时间建立更加具有针对性的免疫应答。机体受到相同的抗原再次刺激后，多数情况下会产生二次应答。由于有了初次应答的记忆，所以二次应答反应更加及时迅速，无须重新学习。应答的基本过程如图 4.6 所示。

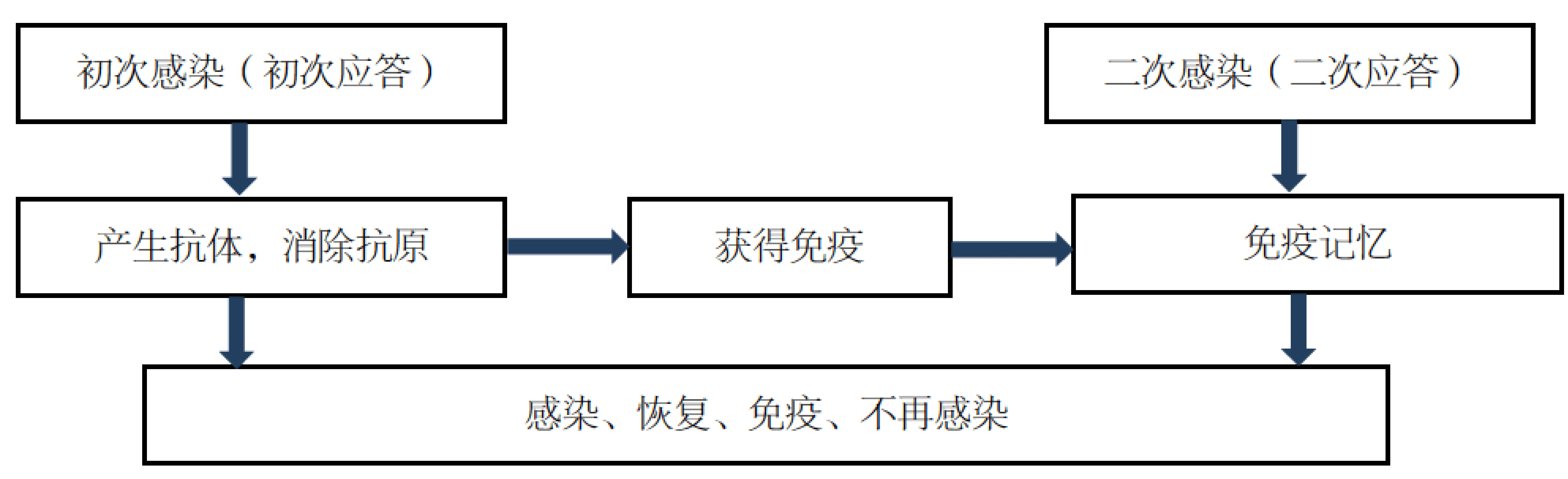


图 4.6　免疫应答的基本过程

免疫系统通过免疫细胞的分裂和分化作用，可产生大量的抗体来抑制各种抗原，具有多样性。免疫系统执行免疫防卫能力的比较细胞为淋巴细胞（包括 T 细胞和 B 细胞），B 细胞的主要作用是识别抗原和分泌抗体，T 细胞能够促进和抑制 B 细胞的产生与分化。当抗原入侵机体后，B 细胞分泌的抗体与抗原发生结合作用，当它们之间的结合力超过一定限度时，分泌这种抗体的 B 细胞将会发生克隆扩增。克隆细胞在其母体的亲和力影响下，按照与母体亲和力成正比的概率对抗体的基因多次重复随机突变及基因块重组，进而产生种类繁多的免疫细胞，并获得大量识别抗原能力比母体强的 B 细胞。这些识别能力较强的细胞能有效缠住入侵抗原，这种现象称为亲和成熟。

一旦有细胞达到最高亲和力，免疫系统就会通过记忆进行大量复制，并直接保留，因而具有记忆功能和克隆能力。B 细胞的一部分克隆个体分化为记忆细胞，再次遇到相同抗原后能够迅速

被激活，实现对抗原的免疫记忆。B 细胞的克隆扩增受 T 细胞的调节，当 B 细胞的浓度增加到一定程度时，T 细胞对 B 细胞产生抑制作用，从而防止 B 细胞的无限复制。当有新的抗原入侵或某些抗体大量复制而被破坏免疫平衡时，通过免疫系统的调节，可以抑制尝试过高或相近的抗体的再生能力，并实施精细进化达到重新平衡，因而具有自我调节的能力。

除了机体本身的免疫功能，可以人为地接种疫苗，起到免疫的作用。疫苗是将细菌、病毒等病原体微生物及其代谢产物，经过人工减毒、灭活或利用基因工程的方法制备的用于预防传染病的自动免疫制剂。疫苗保留了病原菌刺激动物免疫系统的特性，当动物体接触到这种不具有伤害力的病原菌后，免疫系统便会产生一定的保护物质，如免疫激素、活性物质、特殊抗体组织的。当动物再次接触到这种病原菌时，动物体的免疫系统便会依循其原有的记忆，制造更多的保护物质来阻止病原菌的伤害。

4.8.3 人工免疫算法的基本概念

1. 人工免疫系统的定义

目前关于人工免疫系统的定义已经有多种表述，以下是几种比较贴切的定义：

（1）De Castro 给出的第二个人工免疫系统定义：人工免疫系统是受生物免疫系统启发而来的用于求解问题的适应性系统。

（2）Timmis 给出的第二个人工免疫系统定义：人工免疫系统是一种由理论生物学启发而来的计算范式，借鉴了一些免疫系统的功能、原理和模型并用于复杂问题的解决。

（3）国内学者给出的人工免疫系统的定义：人工免疫系统是基于免疫系统机制和理论免疫学而发展的各种人工范例的特称。

生物世界为计算问题求解提供了许多灵感和源泉。人工免疫系统作为一种智能计算方法，它与人工神经网络、进化计算及群集智能一样，都属于基于生物隐喻的仿生计算方法，且都来源于自然界中的生物信息处理机制的启发，并用于构造能够适应环境变化的智能信息处理系统，即是现代信息科学与生命科学相互交叉渗透的研究领域。

2. 免疫算法的基本思想

人工免疫算法主要包括以下几个关键步骤。

（1）产生初始群体。对初始应答，初始抗体随机产生；而对再次应答，则借助于免疫机制的记忆功能，部分初始抗体由记忆单元获取。由于记忆单元中抗体具有较高的适应度和较好的群体分布，因此可提高收敛速度。

（2）根据先验知识抽取疫苗。

（3）计算抗体适应度。

（4）收敛判断。

若当前种群中包含最佳个体或达到最大进化代数，则算法结束，否则进行以下步骤。

（5）产生新的抗体。每一代新抗体主要通过以下两条途径产生。

① 基于遗传操作生成新抗体。采用赌轮盘选择机制，当群体相似度小于阈值时，多样性满

足要求，则抗体被选中的概率正比于适应度，反之，按下述（2）的方式产生新抗体，交叉和变异算子均采用单点方式。

② 随机产生 P 个新抗体。为保证抗体多样性，模仿免疫系统细胞的新陈代谢功能，随机产生 P 个新抗体，使抗体总数为 $N+P$，再根据群体更新，产生规模为 N 的下一代群体。

③ 群体更新。对种群进行接种疫苗和免疫选择操作，得到新一代规模为 N 的父代种群，返回（3）。

免疫算法的流程图如图 4.7 所示。

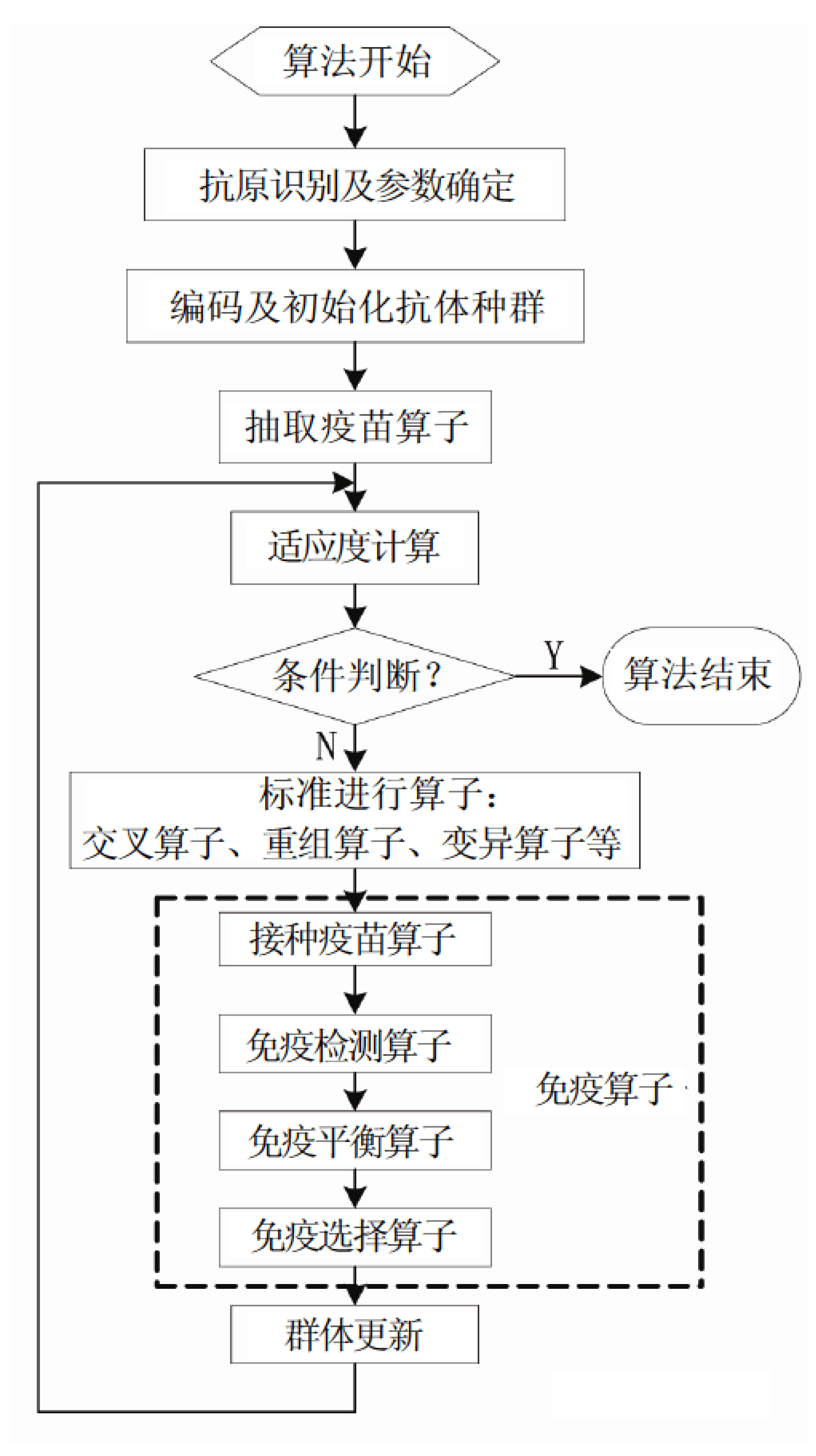


图 4.7 免疫算法的流程图

4.8.4 免疫算子

免疫算子通常包括多种免疫算子：提取疫苗算子、接种疫苗算子、免疫检测算子、免疫平衡算子、免疫选择算子、克隆算子等。增加免疫算子可以提高进化算法的整体性能并使其有选择、有目的地利用特征信息来抑制优化过程中的退化现象。

1. 提取疫苗算子

疫苗是依据人们对待求问题所具备的或多或少的先验知识，它所包含的信息量及其准确性对

算法的运行效率和整体性能起着重要的作用。

首先对所求解的问题进行具体分析，从中提取出最基本的特征信息，然后，对此特征信息进行处理，以将其转化为求解问题的一种方案，最后，将此方案以适当的形式转化为免疫算子，以实施具体的操作。例如在求解 TSP 问题时，可以依据不同城市之间的距离作为疫苗；在应用于模式识别的分类与聚类时，可以依据样品与模板之间或样品与样品之间的特征值距离作为疫苗。由于每一个疫苗都是利用局部信息来探求全局最优解，即估计该在某一分量上的模式，所以没有必要对每个疫苗做到精确无误。如果为了精确，可以尽量将原问题局域化处理得更彻底，局部条件下的求解规律就会越明显。但是这使得寻找这种疫苗的计算量会显著增加。还可以将每一代的最优解作为疫苗，动态地建立疫苗库，当前的最优解比疫苗库中的最差疫苗的亲和力高时，则取代该最差疫苗。

值得提出的是，由于待求问题的特征信息往往不止一个，所以疫苗也可能不止一个，在接种过程中可以随机地选取一种疫苗进行接种，也可以将多个疫苗按照一定的逻辑关系进行组合后再予以接种。

2. 接种疫苗算子

接种疫苗主要是为了提高适应度，利用疫苗所蕴含的指导问题求解的启发式信息，对问题的解进行局部调整，使得候选解的质量得到明显改善。接种疫苗有助于克服个体的退化现象和有效地处理约束条件，从而可以加快优化解的搜索速度，进一步提高优化计算效率。

设个体 x，接种疫苗是指按照先验知识来修改 x 的某些基因位上的基因或其分量，使所得个体以较大的概率具有更高的适应度。这一操作应满足两点：

①若个体 y 的每一基因位上的信息都是错误的，即每一位码都与最佳个体不同，则对任何一个体 x，转移为 y 的概率为 0；

②若个体 x 的每个基因位都是正确的，即 x 已经是最佳个体，则 x 以概率 1 转移为 x。设群体 $c=(x_1,x_2,\cdots,x_n)$ 对 c 接种疫苗是指在 c 中按比例 α 随机抽取 $n_\alpha=\alpha_n$ 个个体而进行的操作。

3. 免疫检测算子

免疫检测是指对接种了疫苗的个体进行检测，若其适应度仍不如父代，说明在交叉、变异的过程中出现了严重的退化现象，这时该个体将被父代中所对应的个体所取代，否则原来的个体直接成为下一代的父代。

4. 免疫平衡算子

免疫平衡算子是对抗体中浓度过高的抗体进行抑制，而对浓度相对较低的抗体进行促进的操作。在群体更新中，由于适应度高的抗体的选择概率高，因此浓度逐渐提高，这样会使种群中的多样性降低。因此某抗体的浓度达到一定值时，就抑制这种抗体的产生；反之，则相应提高浓度低的抗体的产生和选择概率。这种算子保证了抗体群体更新中的抗体多样性，在一定程度上避免了早熟收敛。

（1）浓度计算。

对于每一个抗体，统计种群中适应度值与其相近的抗体的数目，则浓度为

$$c_i = \frac{与抗体i具有最大亲和力的抗体数}{抗体总数}$$

（2）浓度概率计算。

设定一个浓度阈值 T，统计浓度高于该阈值的抗体，记数量为 HighNum。规定这 HighNum 个浓度较高的抗体浓度概率为

$$P_{density} = \frac{1}{抗体总数} \leqslant \left(1 - \frac{\text{HighNum}}{抗体总数}\right)$$

其余浓度较低的抗体浓度概率为

$$P_{density} = \frac{1}{抗体总数}\left(1 + \frac{\text{HighNum}}{抗体总数} \cdot \frac{\text{HighNum}}{抗体总数 - \text{HighNum}}\right)$$

5. 免疫选择算子

免疫选择算子是对经过免疫检测后的抗体种群，依据适应度和抗体浓度确定的选择概率选择出个体，组成下一代种群。

概率的计算公式

$$P_{choose} = \alpha \bullet p_f + (1-\alpha) \bullet p_d$$

式中：p_f 为抗体的适应度概率，定义为抗体的适应度值与浓度值和之比；p_d 为抗体的浓度概率，抗体的浓度越高越会受到抑制，浓度越低则越会受到促进；α 为比例系数，决定了适应度与浓度的作用大小。

然后再利用赌轮盘选择方式，依据计算出的选择概率对抗体进行选择，选出相对适应度较高的抗体作为下一代的种群抗体。

6. 克隆算子

克隆算子源于对生物具有的免疫克隆选择机理的模仿和借鉴。在抗体克隆选择学说中，当抗体侵入机体中，克隆选择机制在机体内选择出识别和消灭相应抗原的免疫细胞，使之激活、分化和增殖，进行免疫应答以最终消除抗原。免疫克隆的实质是在一代进行中，在候选解的附近，根据亲和度的大小，产生一个变异解的群体，扩大了搜索范围，避免了遗传算法对初始种群敏感、容易出现早熟和搜索限于局部极小值的现象，具有较强的全局搜索能力。该算子在保证收敛速度的同时又能维持抗体的多样性。

通过不同的免疫算子和进化算子（交叉算子、重组算子、变异算子和选择算子）的重组融合，可形成不同的免疫进化算法。其中免疫算子可以优化其他智能算法，不仅保留了原来智能算法的优点，同时也弥补了原算法的一些不足和缺点。

4.8.5　免疫算法与免疫系统的对应

免疫算法是借鉴了免疫系统学习性、自适应性以及记忆机制等特点而发展起来的一种优化组合方法，在使用免疫算法解决实际问题时，各个步骤都与免疫系统有对应关系。表 4.2 为免疫算法与免疫系统对应关系表。其中根据疫苗修正个体基因的过程即为接种疫苗，其目的是消除抗原在新个体产生时带来负面影响。

表 4.2　免疫算法与免疫系统对应关系

免疫系统	免疫算法
抗原	要解决的问题
抗体	最佳解向量
抗原识别	问题分析
从记忆细胞产生抗体	联想过去的成功解
淋巴细胞分化	优良解（记忆）的复制保留
细胞抑制	剩余候选解消除
抗体增加（细胞克隆）	利用免疫算子产生新抗体
亲和力	适应度
疫苗	含有解决问题的关键信息

4.8.6　人工免疫算法与遗传算法的比较

人工免疫算法作为一种进化算法，所用的遗传结构与遗传算法中的类似，采用重组、变异等算子操作解决抗体优化问题，但也存在区别：

（1）人工免疫算法起源于抗原与抗体之间的内部竞争，其相互作用的环境包括内部及外部环境；而遗传算法起源于个体和自私基因之间的外部竞争。

（2）人工免疫算法假设免疫元素互相作用，即每一个免疫细胞等个体可以互相作用，而遗传算法不考虑个体间的作用。

（3）人工免疫算法中，基因可以由个体自己选择，而在遗传算法中基因由环境选择。

（4）人工免疫算法中，基因组合是为了获得多样性，一般不用交叉算子，因为人工免疫算法中基因是在同一代个体进行进化，这种情况下，设交叉概率为 0；而遗传算法后代个体基因通常是父代交叉的结果，交叉用于混合基因。

（5）人工免疫算法选择和变异阶段明显不同，而遗传算法中它们是交替进行的。

所以，也可以把人工免疫算法看作是遗传算法的补充。

与遗传算法相比，人工免疫算法在个体更新、选择算子、维持多样性等方面有很大的改进。

（1）个体更新。在遗传算法中的交叉、变异算子之后，人工免疫算法利用先验知识，引入疫苗接种算子，这样对随机选出的个体的某些基因位，用疫苗的信息来替换，从而使个体向最优解逼近，加快了算法的收敛速度，实现个体更新的过程。

（2）选择算子。在遗传算法中，在个体更新后并没有判断其是否得到了优化，以至于经过交叉、变异后的个体不如父代个体，即出现退化现象。而在人工免疫算法中，在经过交叉、变异、疫苗接种算子的作用后，新生成的个体需要经过免疫检测算子操作，即判断其适应度是否优于父代个体，如果发生了退化，则用父代个体替换新生成个体，然后利用抗体的适应度值和浓度值所共同确定的选择概率，参加轮盘赌选择操作，最终选择出新一代种群。

（3）维持多样性。在遗传算法中，适应度高的个体在一代中被选择的概率高，相应的浓度高；适应度低的个体在一代中被选择的概率低，相应的浓度低，没有自我调节功能。而在人工免疫算法中，除了抗体的适应度，还引入了免疫平衡算子参与到抗体的选择中。免疫平衡算子对浓度高的抗体进行抑制，反之对浓度低的抗体进行促进。由于免疫平衡算子的引入，使得抗体与抗

体之间的相互促进或抑制，维持了抗体的多样性及免疫平衡，体现了免疫系统的自我调节功能。

正是存在着与遗传算法不同的特点，人工免疫算法具有分布式、并行性、自学习、自适应、自组织、鲁棒性和凸显性等特点。与传统数学方法相比，人工免疫算法在进行问题求解时，与进化计算方法相似，不需依赖于问题本身的严格数学性质，如连续性和可导性等，不需要建立关于问题本身的精确数学描述，一般也不依赖于知识表示，而是在信号或数据层直接对输入信号进行处理，属于求解那些难以有效建立形式化模型、使用传统方法难以解决或根本不能解决的问题。人工免疫算法是一种随机概率型的搜索方法，这种不确定性使其能有更多的机会求得全局最优解；人工免疫算法又是利用概率搜索来指导其搜索方向，概率被作为一种信息来引导搜索过程朝搜索空间更优化的解区域移动，有着明确的搜索方向，算法具有潜在的并行性，并且易于并行化。

4.9　基于 MATLAB 的进化算法

进化算法的 MATLAB 实现，除了自己编写程序外，还可以采用 MATLAB 中的遗传算法。使用此工具箱，可以扩展 MATLAB 及其优化工具箱在处理优化问题方面的能力，可以处理传统的优化技术难以解决的如难以定义或不便于进行数学建模的问题；也可以解决目标函数较复杂的问题，比如目标函数不连续或具有高度非线性、随机性以及目标函数不可微等。

在 MATLAB 中，遗传算法工具箱中的函数可以通过命令行和图形用户界面（GUI）两种方式来调用。在使用图形用户界面时，通过相应窗格进行遗传算法的各参数的设置及计算。在用命令行实现遗传算法时，则通过调用相应的遗传算法函数进行算法设置并完成计算。

需要注意的是，遗传算法工具箱中的优化函数总是使目标函数最小化，如果想要求出函数的最大值，可以转化求取函数的负函数的最小值。

例 2.11　体重约 70kg 的某人在短时间内喝下 2 瓶啤酒后，隔一段时间测量他的血液中酒精含量（mg/100mL），得到表 4.3 的数据。

表 4.3　酒精在人体血液中分解的动力学数据

时间/(h)	0.25	0.5	0.75	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
酒精含量/	30	68	75	82	82	77	68	68	58	51	50	41
时间/(h)	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	
酒精含量	38	35	28	25	18	15	12	10	7	7	4	

根据酒精在人体血液分解的动力学规律可知，血液中酒精浓度与时间的关系可表示为

$$c(t)=k(e^{-qt}-e^{-rt})$$

试根据表中数据求出参数 k、q、r。

解：

利用 MATLAB 的遗传算法工具箱的两种方法求解此问题。首先编写目标函数并以文件名 myfun 存盘：

```
function y=myfun(x)
c=[30 68 75 82 82 77 68 68 58 51 50 41 38 35 28 25 18 15 12 10 7 7 4];
t=[0.25 0.5 0.75 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 6.0 7.0 8.0 9.0 10.0 11.0
12.0 13.0 14.0 15.0 16.0];
```

```
[r,s]=size(c);y=0;
for i=1:s
   y=y+(c(i)-x(1)*(exp(-x(2)*t(i))-exp(-x(3)*t(i))))^2;      %残差的平方和
end
```

1. 命令行方法

在 MATLAB 工作窗口输入下列命令：

```
>>Lb=[-1000, -10, -10];                         %定义下界
>>Lu=[1000,10,10];                              %定义上界
>>x_min=ga(@myfun,3,[],[],[],[],Lb,Lu)          %要注意 myfun 函数应在 MATLAB 搜索途径上
   得到结果：x_min =72.9706    0.0943    3.9407
```

由于遗传算法是一种随机性的搜索方法，所以每次运算可得到不同的结果。为了得到最终的结果，用直接搜索工具箱中的 fminsearch 函数进行验证：

```
>> fminsearch(@myfun,x_min)          %利用遗传算法得到的值作为搜索初值，以减少搜索时间
   ans =114.4325    0.1855    2.0079      %最终结果
```

图 4.8 为用原始数据及用优化结果所绘制的曲线。

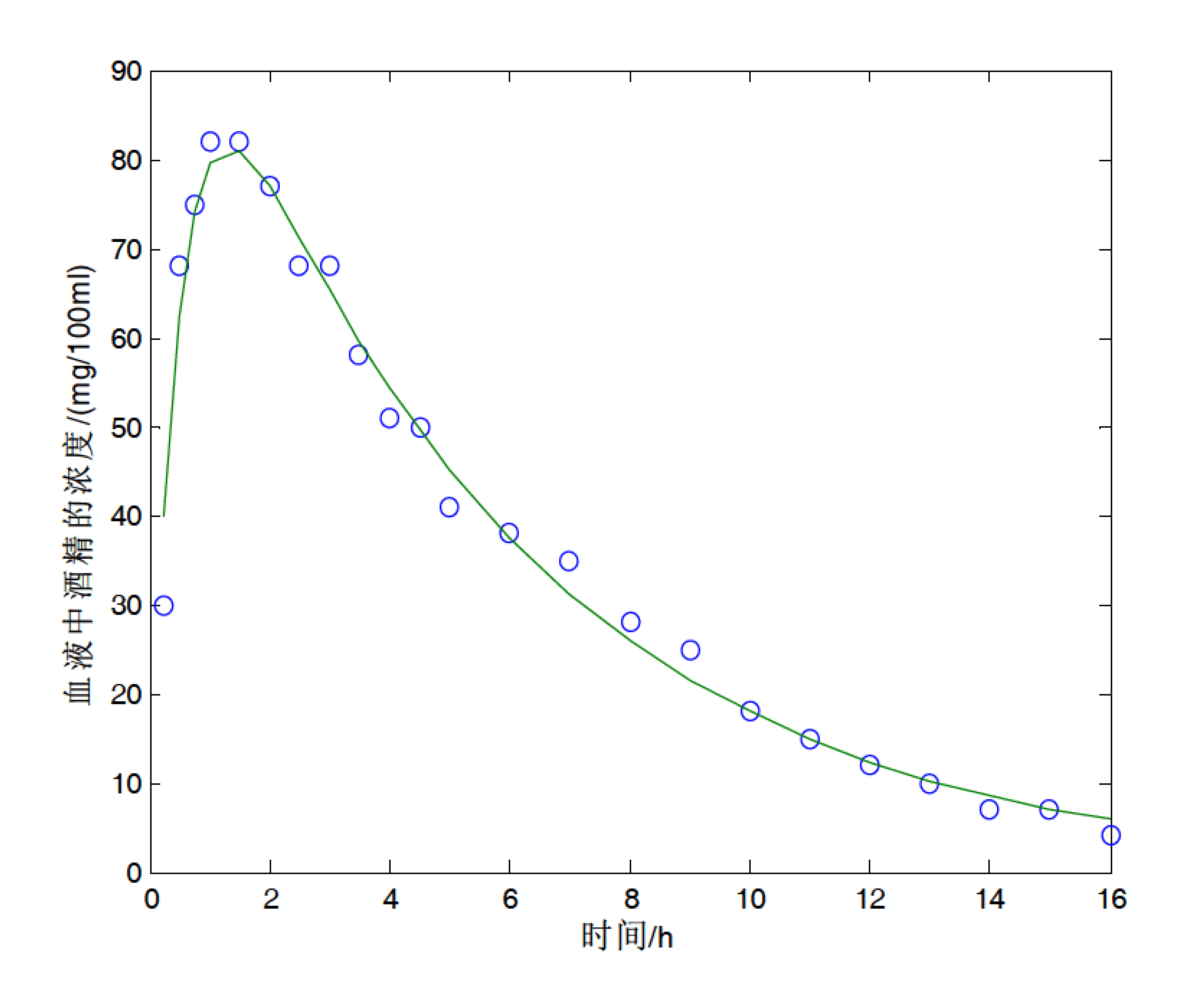


图 4.8　酒精在人体血液中分解的动力学曲线

从这个例子可看出，用遗传算法求解非线性最小二乘问题时，对最终的结果要用其他方法进行验证。

2. GUI 方法

在较高的 MATLAB 版本中，遗传算法工具箱等各种优化方法都包含在 optimtool（优化用户图形界面）。在工作窗口输入：

```
>>optimtool
```

打开 GUI，在 Solver 窗口选择 Ga-gnetic Algorithm，在 Problem 框架内的 Fitness Function 窗口中输入@myfun，在 number of variables 输入变量数目 3；在 Constraints（约束条件选择框）的 Bounds 的 Lower 窗口中输入[−1000,−10,−10]。在 Upper 窗口输入[1000, 10, 10]。在 Options（优化参数）中的所有参数选缺省值。

然后单击 Start 运行遗传算法，其中一次的结果为：36.368　0.036　9.984

例 2.12　求下列函数的最优值：

$$f(x,y)=x\sin(4\pi x)-y\sin(4\pi y+\pi+1),\qquad x,y\in[-1,2]$$

解：

此函数的图形见图 4.9，有极大值为 f（1.6289,2）=3.3099。

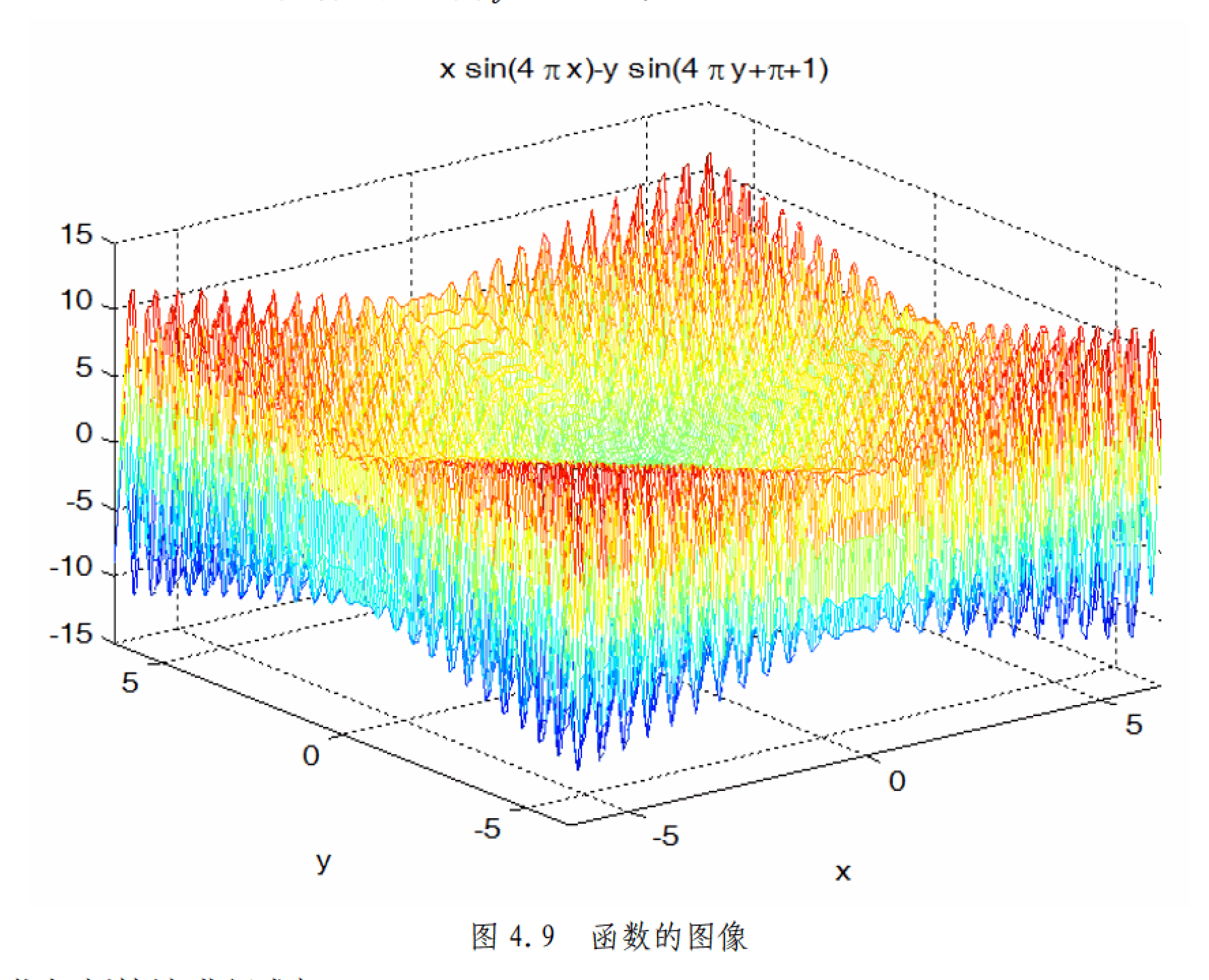


图 4.9　函数的图像

用进化规划算法进行求解：

```
>>zfun=inline('x*sin(4*pi*x)-y*sin(4*pi*y+pi+1)'); ezmesh(zfun,100)    %作函数图像
>>myvar=[300 400 100];c_best=myEP(myvar);
>> c_best
  c_best =x: 1.6289  y: 2  fitness: 3.3099
```

例 2.13　利用进化策略算法对渐进回归模型 $f(x)=\alpha-\beta\gamma^x$ 的参数进行估计，实验估计如表 4.4 所示。

表 4.4　实验数据

x	12	23	40	92	156	215
y	0.094	0.119	0.199	0.260	0.309	0.331

解：

根据进化策略算法的原理，可编程计算得到以下的结果（zuijie）：

```
   α=0.2734  β=0.2336  γ=0.9788
>>u=50;lenda=350;a=2*rand(u,1);b=rand(u,1);c=rand(u,1);A=zeros(u,3);
A=[a b c];sigma=0.5*ones(u,3);
x=[12 23 40 92 156 215];y=[0.094 0.119 0.199 0.260 0.309 0.331];
y1=y(ones(u,1),:);x1=x(ones(u,1),:);f1=zeros(u,6);
for i=1:u                                              %适应度函数计算
   for j=1:6;f1(i,j)=A(i,1)-A(i,2)*A(i,3)^x1(i,j);end
end
g=zeros(u,1);g=sum((f1-y1).^2,2);
[g,index]=sort(g);zuijie(1,:)=A(index(1),:);jie=g(1);      %最优解
t=0;AA=zeros(lenda,3);sigma1=zeros(lenda,3);
while t<500
if jie<1e-6;break;end
   for k=1:lenda                                       %混合重组
k1=randperm(u); AA(k,:)=(A(k1(1),:)+A(k1(2),:))./2;
     sigma1(k,:)=(sigma(k1(1),:)+sigma(k1(2),:))./2;
end

r=1;r1=1;ra=randn(lenda,3);ra1=randn(lenda,3);sigma1=sigma1.*exp(r1*ra1+r*ra);
   AA=AA+sigma1.*ra;                                   %高斯变异
   for i=1:lenda                                       %边界处理
      if AA(i,1)>2||AA(i,1)<0;AA(i,1)=2*rand;end
      if AA(i,2)>1||AA(i,2)<0;AA(i,2)=rand;end
      if AA(i,3)>1||AA(i,3)<0;AA(i,3)=rand;end
end
   yy1=y(ones(lenda,1),:);
xx1=x(ones(lenda,1),:);G=zeros(lenda,1);ff1=zeros(lenda,6);
for i=1:lenda
      for j=1:6;ff1(i,j)=AA(i,1)-AA(i,2)*AA(i,3)^xx1(i,j);end
```

```
end
    G=sum((ff1-yy1).^2,2);
[G,index]=sort(G);zuijie(1,:)=AA(index(1),:);jie=G(1);
    A=AA(index(1:u),:);sigma=sigma1(index(1:u),:);%(u,λ)策略
    t=t+1;
end
```

例 2.14　拟对陕西省进行喷灌区划，其一级区预分 3 类。从陕南、关中、陕北地区选择 27 种作物作为样本，数据如表 4.5（各变量代表的物理意义及作物名称从略）。试用基于动态疫苗提取的疫苗遗传算法对其进行分类。

表 4.5　原始数据

样本编号	地　区	X_1	X_2	X_3
1	陕南	45	0.2	1903
2		250	10.88	208.92
3		225	19.2	146.92
4		49.6	7.75	146.05
5		240	26.4	6.25
6		220	26.4	223.1
7		240	26.4	203.1
8		16.5	12.29	−17.29
9		20.5	6.91	−5.41
10	关中	22.71	3.0	0.71
11		36.68	5.2	15.48
12		97.85	3.0	68.85
13		240	39.6	219.9
14		220	39.6	189.9
15		240	39.6	209.9
16		110	4.95	67.05
17	陕北	11.82	5.2	−2.91
18		12.38	5.2	−2.41
19		6.78	5.2	−8.00
20		21.9	5.2	7.12
21		9.35	5.2	−2.80
22		14.7	4.4	−13.70
23		8.48	5.2	−3.66
24		132	5.2	92.72
25		107.2	5.2	65.42
26		130.0	8.25	127.25
27		120.0	8.25	117.75

解：

所谓动态疫苗是指建立动态的疫苗库（数量可以固定，也可以是变化的），将每一代的最优个体放入疫苗库中，在每次向疫苗库中加入新的疫苗后，都要按适应度对疫苗进行排序，淘汰适应度较小的疫苗。接种时随机选取动态疫苗库的疫苗，对子代种群进行接种，随机指定某位基因，依据选择疫苗中相应的基因来修改抗体对应基因位上的值。

疫苗遗传算法求解聚类时，随机指定各样本的类号，再进行交叉、变异、接种等操作，得到新的分类方式，并根据适应度的变化确定最佳的分类方式。

据此，可以编程计算并得到以下的结果（其中的一次结果）：

```
>>a=dyICA(myvar)
pattern: [1  1  1  2  1  1  2  1  1  1  3  1  1  3  3  1  1  3  2  3  1  1  1  1  3  2  2]
    fitness: 37.8777
```

例 2.15　利用 Memetic（基因）算法求解下列的 TSP 问题。其中城市坐标如表 4.6 所示。

表 4.6　各城市的坐标

城市	1	2	3	4	5	6	7
X	16.47	16.47	20.09	22.39	25.23	22.00	20.47
Y	96.10	94.44	92.54	93.37	97.24	96.05	97.02
城市	8	9	10	11	12	13	14
X	17.20	16.30	14.05	16.53	21.52	19.41	20.09
Y	96.29	97.38	98.12	97.38	95.59	97.13	94.55

解：

为了提高算法全局极值的搜索能力，本题对算法中的交叉算子作如下改进：随机选择两个个体，并进行交叉，产生一个后代，对其进行局部优化；对后代和当前最优个体进行交叉，产生一个后代，对其进行局部优化。

据此，可编程计算，得到以下结果。图 4.10 为该问题的解。从运行结果分析，通过这样的改进，可以很快地找到最优点。

```
   >>myval=[80 3 0.6 0.05 300];
   >>popsize=myval(1);searchnum=myval(2);pc=myval(3);pm=myval(4);iter_max=my
val;
   >> [y,x]=memetic_TSP(popsize,searchnum,pc,pm,iter_max)
   y=30.8013    x=12－6－5－4－3－14－2－1－10－9－11－8－13－7
```

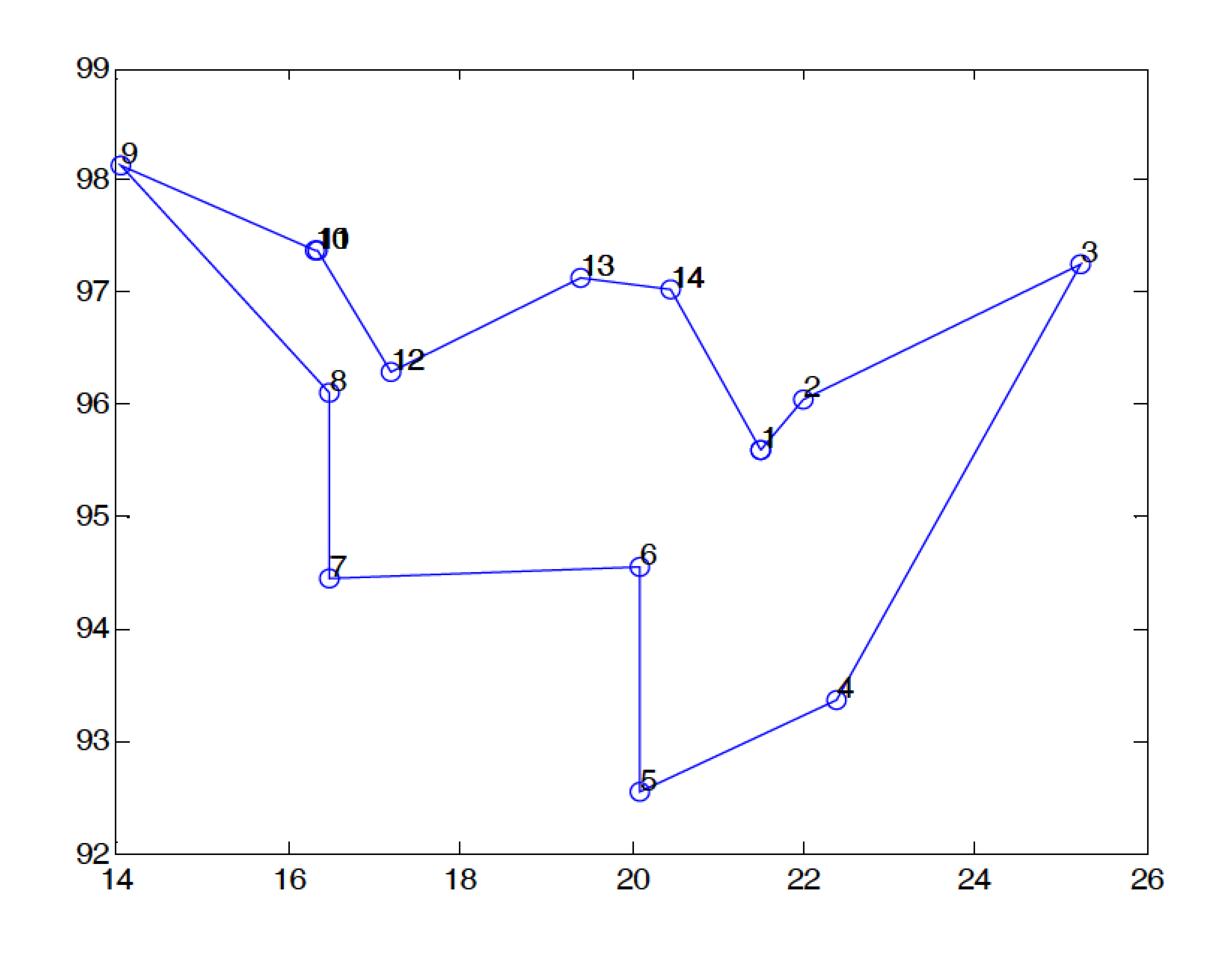


图 4.10　问题的解

例 2.16　利用量子遗传算法求解下列函数的极值：

$$\min f(x_i) = -20e^{-0.2\sqrt{\frac{x_1^2+x_2^2}{2}}} - e^{\frac{1}{2}(\cos(2\pi x_1)+\cos(2\pi x_2))} + 22.71282 \quad |x_i| \leqslant 5$$

解：

此函数是一个多峰函数，其全局有一极小值：

$$f(0,0)=0$$

根据量子遗传算法的原理，可编程计算得到以下的结果：

```
>>myval=[20 0.05 2000 16];
>>popsize=myval(1);pm=myval(2);iter_max=myval(3);num=myval(4);
>> [y_max,x_max]=QGA(popsize,pm,iter_max,num)
y_max =0.0207    x_max=-0.0114  0.0002
```

例 2.17　利用人工免疫算法对下列函数寻优：

$$\max \quad f(x,y) = \left(\frac{3}{0.05+(x^2+y^2)}\right)^2 + (x^2+y^2)^2 \quad -5.12 \leqslant x,y \leqslant 5.12$$

解：

此函数的图形见如图 4.11 所示，在（0,0）处有极大值 3600。

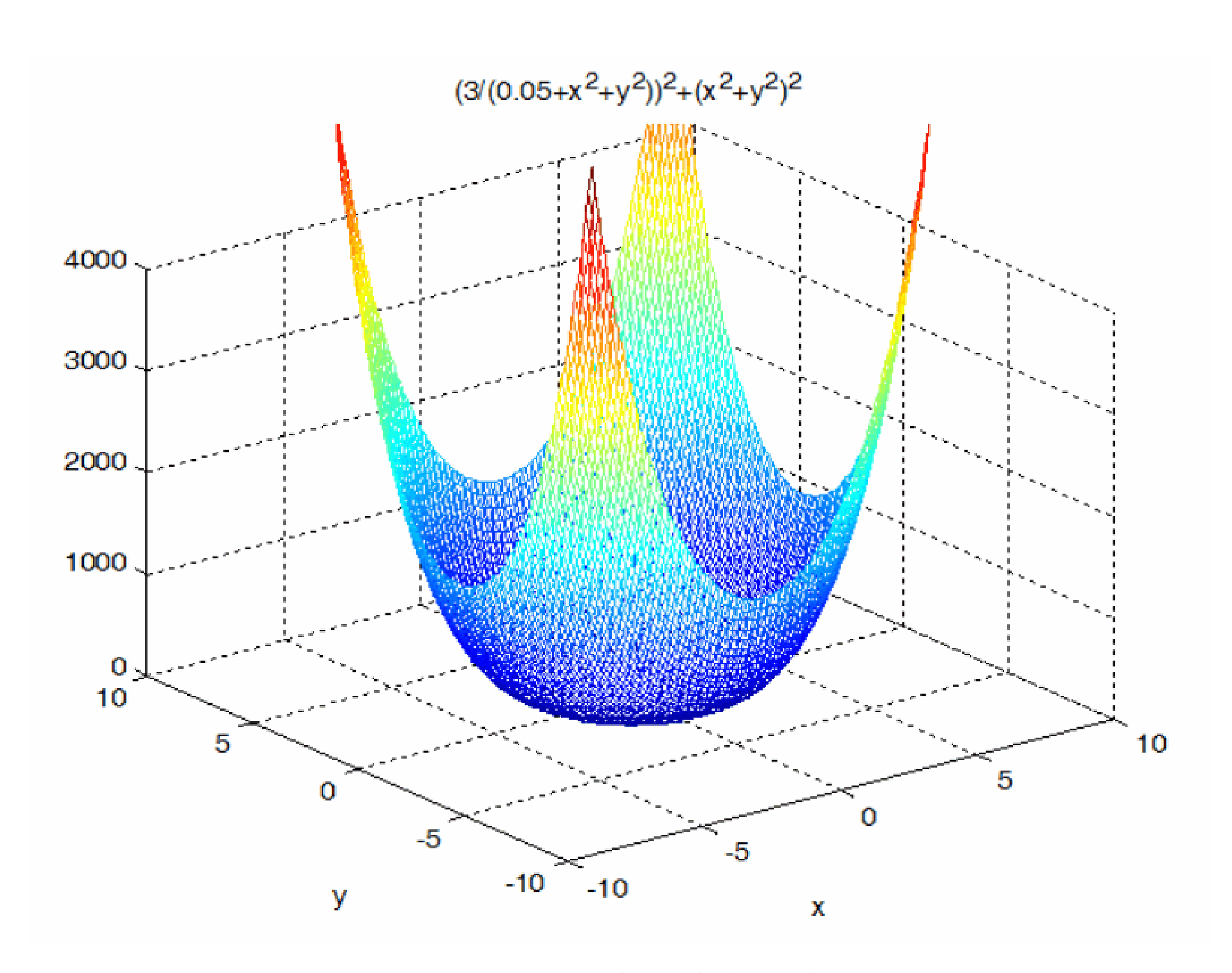


图 4.11 所求函数的图形

在使用人工免疫算法时要解决以下问题：

（1）抗体个体的编码：虽然可以采用二进制编码，其搜索能力较强但需要频繁进行交换的编码与解码，计算工作量大且只能产生有限的离散值，所以在此采用十进制（实数编码），它利用如下线性变换进行编码：

$$x_j = a_j + u_0(j)(b_j - a_j)$$

把初始变化区间$[a_j,b_j]$第 j 个优化变量 x_j映射到[0,1]区间上的实数 $u0$(j)，即基因编码。

（2）抗体浓度的计算：在计算中一般根据以下的标准判断抗体的相似性：

$$\frac{f_i}{f_j} \leqslant 1+\varepsilon$$

其中：ε 为一个较小的正数，如为 0.02 表示抗体 i 与抗体 j 之间的相似度有 98%。

（3）疫苗的建立及接种：不同的问题可能有不同的疫苗，所以要根据具体的先验知识来确定疫苗。在此为了使其算法具有更加的通用性，根据以下方法建立疫苗：

① 建立疫苗库：一般将数目为 20%～40%群体规模的第 k−1 代迭代过程中所产生的较优抗体作为疫苗库。

② 根据轮盘赌选择策略从疫苗库中选择出某较优的个体作为疫苗。

③ 将疫苗接种于选择的个体，此时可以将疫苗全部替换被选择个体基因位，也可以替换部分基因位。

根据人工免疫算法的原理，编程进行计算，并得到以下的结果（其中一次的结果）：

```
>>zfun=inline('(3/(0.05+x^2+y^2))^2+(x^2+y^2)^2'); ezmesh(zfun,100);
>>axis([-10 10 -10 10 0 4000]);myvar=[300 0.4 0.05 300 2];
>>a=myICA(myvar)
  var: [5.6320e-004 -7.6800e-004]    fitness: 3.5999e+003
```

第 5 章

统计分析方法

统计分析涉及数据收集、描述及分析推理等步骤，虽然从传统意义上讲，统计分析方法不是数据挖掘，但很多统计概念是数据挖掘技术的基础，在解决数据挖掘问题时，有时会先使用统计方法去试着解决问题，或者用统计分析方法进行数据预处理。

假设检验、回归分析以及方差分析是经典统计学中的主要内容。假设检验是一种用于“证实”某种假设或论断的方法；回归分析是探索研究对象的模型和预测未知特征的方法；方差分析是判断不同因素之间差异的方法，它将所有差异分解成系统差异和随机差异。

5.1 假设检验

假设检验中有二类重要问题，第一个是根据样本的信息判断总体分布是否具有指定的特征；第二个是在估计某未知参数 β 时，除了求出它的点估计外，还希望在一定的置信水平上估计出一个范围，即 β 的置信区间。

5.1.1 随机误差的判断

随机误差的大小可用试验数据的精密度来反映，而精密度的高低可用方差来量度，所以对测试结果进行方差检验，即可判断各试验方法或试验结果的随机误差之间的关系。

1. χ^2 检验

χ^2 检验适用于单个正态总体的方差检验，即在试验数据的总体方差已知的情况下，对试验数据的随机误差或精密度进行检验。

假设有一组数据 $x_1,x_2,\cdots,x_n$ 服从正态分布，则统计量

$$\chi^2=\frac{n-1}{\sigma_0^2}s^2\sim\chi_{(n-1)}^2$$

对于给定的显著性水平，可与由相应的 χ^2 分布表查得的临界值进行比较，就可判断两方差之间有无显著差异。显著性水平 α 一般为 0.01 和 0.05。

双尾检验时，若 $\chi^2_{1-\frac{\alpha}{2}}<\chi_0^2<\chi^2_{\frac{\alpha}{2}}$，则可判断该组数据的方差与原总体方差无显著差异，否则有显著差异，并且标准差 σ^2 在 $1-\alpha$ 水平上的置信区间为 $\left[\frac{\frac{n-1}{}s^2}{\chi^2_{\frac{\alpha}{2},n-1}},\frac{\frac{n-1}{}s^2}{\chi^2_{1-\frac{\alpha}{2},n-1}}\right]$。

单尾检验时，若 $\chi_0^2>\chi^2_{1-\alpha,n-1}$，则判定该组数据的方差与原总体方差无显著性减小，否则有显著减小，并且标准差 σ^2 在 $1-\alpha$ 水平上的置信区间为 $\left[-\infty,\frac{\frac{n-1}{}s^2}{\chi^2_{1-\alpha,n-1}}\right]$。此为左尾检验。

若 $\chi_0^2<\chi^2_{\alpha,n-1}$，则判定该组数据的方差与原总体方差无显著增大，否则有显著增大，并且标准差 σ^2 在 $1-\alpha$ 水平上的置信区间为 $\left[\frac{\frac{n-1}{}s^2}{\chi^2_{\alpha,n-1}},+\infty\right]$。此为右尾检验。

如果对所研究的问题只需判断有无显著差异，则采用双尾检验；如果所关心的是某个参数是否比某个值偏大（或偏小），则宜采用单尾检验。

2. *F*检验

*F*检验适用于两组具有正态分布的试验数据间的精密度的比较。

设有两组试验数据 $x_1,x_2,\cdots,x_{n1}$ 与 $y_1,y_2,\cdots,y_{n2}$，两组数据都服从正态分布，样本方差分别为 s_1^2 和 s_2^2，则统计量

$$F=\frac{s_1^2}{s_2^2}\sim F(n_1-1,n_2-1)$$

对于给定的检验水平 α，将所计算的统计量 F 与查表得到的临界值比较，即可得出检验结论。

双尾检验时，若 $F_{1-\frac{\alpha}{2}}(n_1-1,n_2-1)<F<F_{\frac{\alpha}{2}}(n_1-1,n_2-1)$，表示 s_1^2 和 s_2^2 无显著性差异，否则有显著差异。

单尾检验时，若 $F<1$，且 $F>F_{1-\alpha}(n_1-1,n_2-1)$，则可判断 s_1^2 比 s_2^2 无显著减小，否则有显著减小，此为左尾检验；若 $F>1$，且 $F<F_{1-\alpha}(n_1-1,n_2-1)$，则可判断 s_1^2 比 s_2^2 无显著性增大，否则有显著增大，此为右尾检验。

5.1.2　系统误差的检验

在相同条件下的多次重复试验下不能发现系统误差，只有改变形成条件误差的条件，才能发现系统误差。对系统结果必须进行检验，以便能及时减小或消除系统误差，提高试验结果的正确度。

若试验数据的平均值与真值的差异较大，就认为试验数据的正确性不高，试验数据与试验方法的系统误差较大，所以对实验数据的平均值进行检验，实际上是对系统误差的检验。

1. 平均值与给定值比较

如果有一组试验数据服从正态分布，要检验这组数据的算术平均值是否与给定值有显著差异，则检验统计量

$$t=\frac{\overline{X_n}-\mu_0}{\frac{s}{\sqrt{n}}}\sim t_{n-1}$$

式中：$\overline{X_n}$ 是试验数据的算术平均值；s 是 n（n<30）个各试验数据的样本标准差；μ_0 是给定值（可以是真值、期望或标准值），根据给定的显著性水平 α，将计算的值与临界值比较，即可得到检验结论。

双尾检验时，若 $|t|>t_{\frac{\alpha}{2},n-1}$，则可判断该组数据的平均值与给定值无显著性差异，否则就有显著性差异，并且均值在 $1-\alpha$ 水平上的置信区间为 $\left[\overline{X_n}-t_{\frac{\alpha}{2},n-1}\frac{s}{\sqrt{n}},\overline{X_n}+t_{\frac{\alpha}{2},n-1}\frac{s}{\sqrt{n}}\right]$。

左尾检验时，若 t<0，且 $t>-t_{\alpha,n-1}$ 或 $|t|<t_{\alpha,n-1}$，则可判断该组数据的平均值与给定值无显著性差异，否则有显著减小，并且均值在 $1-\alpha$ 水平上的置信区间为 $\left[-\infty,\overline{X_n}+t_{\alpha,n-1}\frac{s}{\sqrt{n}}\right]$。

右尾检验时，若 $t>0$，且 $t<t_{\alpha,n-1}$，则可判断该组数据的平均值与给定值无显著增大，否则有显著增大，并且均值在 $1-\alpha$ 水平上的置信区间为 $\left[\overline{X_n}-t_{\alpha,n-1}\frac{s}{\sqrt{n}},+\infty\right]$。

2. 两个平均值的比较

设有两组试验数据，$x_1,x_2,\cdots,x_{n1}$ 与 $y_1,y_2,\cdots,y_{n2}$，两组数据都服从正态分布，根据两组数据的方差是否存在显著差异，可分为以下两种情况进行分析。

如果两组数据的方差无显著差异，则统计量

$$t=\frac{\overline{x}-\overline{y}}{s_w\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}\sim t_{n_1+n_2-2}$$

其中：s_w 为合并标准差，其计算公式为

$$s_w=\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$$

如果两组数据的精密度或方差有显著差异，则统计量

$$t=\frac{\overline{x}-\overline{y}}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}}\sim t_{df}$$

其中

$$df=\frac{(s_1^2/n_1+s_2^2/n_2)^2}{\frac{(s_1^2/n_1)^2}{n_1+1}+\frac{(s_2^2/n_2)^2}{n_1+1}}-2$$

根据给定的显著性水平 α，将计算的 t 值与临界值比较，即可得到检验结论。

双尾检验时，若 $|t|<t_{\frac{\alpha}{2}}$，则可判断两平均值无显著差异，否则就有显著差异。

单尾检验（左尾检验）时，若 $t<0$ 且 $t>-t_{\alpha,df}$ 或 $|t|<t_{\alpha,df}$，则可判断平均值 1 与平均值 2 无显著减小，否则有显著减小。

单尾检验（右尾检验）时，若 $t>0$，且 $|t|<t_{\alpha,df}$ 可判断平均值 1 较平均值 2 无显著差异，否则有显著增大。

3. 成对数据的比较

在这种试验中，试验数据是成对出现的，除了被比较的因素之外，其他条件相同。例如用两种分析方法或用两种仪器测定同一来源的样品。

成对数据的，是把成对数据之差的总体平均值与零或其他指定值进行比较，采用的统计量为

$$t = \frac{\overline{d} - d_0}{s_d / \sqrt{n}} \sim t_{n-1}$$

式中：d_0 可取零或给定值；d 是成对测定值之差的算术平均值，即

$$d = \frac{\sum_{i=1}^{n}(x_i - y_i)}{n} = \frac{\sum_{i=1}^{n} d_i}{n}$$

s_d 是 n 对试验值之差的样本标准差，即

$$s_d = \sqrt{\frac{\sum_{i=1}^{n}(d_i - \overline{d})^2}{n-1}}$$

对于给定的显著性水平 α，如果 $|t| < t_{\frac{\alpha}{2}}$，则成对数据之间不存在显著的系统误差，否则两组数据之间存在显著的系统误差。

应注意的是，成对试验的自由度为 $n-1$，而分组试验时的自由度为 n_1+n_2-2，后者自由度较大，所以统计检验的灵敏度较高。

一般来讲，当所研究因素的效应比其他因素的效应大得多，或其他因素可以严格控制时，采用分组试验比较合适，否则可采用成对试验。

5.2　回归分析

在实际应用中，常常要面对不确定的预测问题，如产品销量（或销售额）的预测是一个各企业都关注的、不确定的问题。产品的销量受多种因素变化的影响，包括产品质量、价格、价值、折扣、信誉、品牌、偏好等，也即销量 Y 与影响因素 x_i（$i = 1,2,\cdots,k$）的关系可以表示为

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \varepsilon$$

式中：ε 是除 x_i 外的其他不确定因素。对于这个问题的解决，需借助于回归分析。

回归分析（Regression Analysis）是一种处理变量之间相关关系最常用的统计方法，用它可以寻找隐藏在随机性后面的统计规律，即确定回归方程，并通过检验确定回归方程的可信度。

5.2.1　一元线性回归分析

1. 一元线性回归的数学模型

一元线性回归又称直线拟合，是处理两个变量间关系的最简单模型，其回归模型为

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon \quad (\varepsilon \sim N(0,\sigma^2) \quad Y \sim N(\beta_0 + \beta_1 x_1, \sigma^2))$$

上式表明，因变量 Y 的变化由两部分组成：一部分是由于自变量 x 的变化而引起的线性变化部分；另一部分是由于其他随机因素引起的变化部分，即不确定量 ε。其中 β_0、β_1 称为回归系数。

回归分析就是采用合适的方法求得以下的回归方程，并进行检验。

$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1 x_1$$

通常采用最小二乘法求解回归系数 $\hat{\beta}_0$、$\hat{\beta}_1$，即求解下列最小值问题

$$\min Q(\beta_0,\beta_1)=\sum_{i=1}^{n}[Y_i-(\hat{\beta}_0+\hat{\beta}_1 x_i)]^2$$

可以采用多种方法解这个最优问题，最常用的是微分法。可以得到回归系数的计算公式

$$\begin{cases}\hat{\beta}_0=\overline{Y}-\hat{\beta}_1\overline{x}\\ \hat{\beta}_1=\dfrac{\sum\limits_{i=1}^{n}x_iY_i-\dfrac{1}{n}\sum\limits_{i=1}^{n}x_i\sum\limits_{i=1}^{n}Y_i}{\sum\limits_{i=1}^{n}x_i^2-\dfrac{1}{n}(\sum\limits_{i=1}^{n}x_i)^2}\end{cases}$$

其中：$\overline{x}=\dfrac{1}{n}\sum\limits_{i=1}^{n}x_i,\overline{Y}=\dfrac{1}{n}\sum\limits_{i=1}^{n}Y_i$。

2. 回归方程的显著性检验

通过以上方法得到的回归方程不一定总有意义。因此需要对得到的回归系数做显著性检验。回归系数的显著性检验有多种方法，常用的是以下几种。

（1）F 检验－方差分析。

记：$S_T^2=\sum\limits_{i=1}^{n}(Y_i-\overline{Y})^2$，$S_R^2=\sum\limits_{i=1}^{n}(\hat{y}_i-\overline{Y})^2$，$S_E^2=\sum\limits_{i=1}^{n}(Y_i-\hat{y}_i)^2$

有关系式：$S_T^2=S_R^2+S_E^2$，其中 S_E^2 称为残差平方和，S_R^2 称为回归平方和。

考虑检验假设：H_0：$b=0$；H_1：$b\neq0$，在 H_0 为真时，有

$$F=\frac{S_R^2/1}{S_E^2/(n-2)}=\frac{S_R^2}{S_E^2}\sim F(1,n-2)$$

对给定的显著性水平 α，当 $F\geqslant F_\alpha(1,n-2)$ 时，可以认为 $b=0$ 不真，称方程是显著的；反之，方程为不显著。

通常，若 $F\geqslant F_{0.01}(1,n-2)$，则为高度显著；若 $F_{0.05}(1,n-2)\leqslant F\leqslant F_{0.01}(1,n-2)$，则为显著；若 $F<F_{0.05}(1,n-2)$ 则为不显著。

（2）相关系数 r 检验法。

相关系数 r 是反映变量 X 与 Y 呈线性关系程度的一个量度指标，其取值范围是 $|r|\leqslant1$，当接近于 1 时，表明变量 X 与 Y 密切线性相关；当接近于 0 时，则这两者之间为非线性相关。

样本相关系数的计算公式为

$$r=\frac{\sum\limits_{i=1}^{n}(x_i-\overline{x})(Y_i-\overline{Y})}{\sqrt{\sum\limits_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum\limits_{i=1}^{n}(Y_i-\overline{Y})^2}}$$

通过查表可得，由自由度（$n-2$）及显著性水平决定的相关系数显著性临界值 r_α，若 $|r|\leqslant r_\alpha$，接受原假设 H_0，即相关性不显著，否则在 α 水平上显著。

（3）β_0 与 β_1 的检验。

检验 β_0 的统计量为

$$t=\frac{\beta_0}{\sqrt{\frac{\sum_{i=1}^{n}(Y_i-\hat{y}_i)^2}{n-2}\left(\frac{1}{n}+\frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}\right)}}\sim t(n-2)$$

β_0 的标准差为

$$S_{\beta_0}=\sqrt{\frac{\sum_{i=1}^{n}(Y_i-\hat{y}_i)^2}{n-2}\left(\frac{1}{n}+\frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}\right)}$$

β_0 的（1－α）×100%的置信区间为

$$[\beta_0-t_\alpha S_{\beta_0},\beta_0+t_\alpha S_{\beta_0}]$$

检验 β_1 的统计量为

$$t=\frac{\beta_1}{\sqrt{\frac{\sum_{i=1}^{n}(Y_i-\hat{y}_i)^2}{(n-2)\sum_{i=1}^{n}(x_i-\bar{x})^2}}}\sim t(n-2)$$

β_1 的标准差为

$$S_b=\sqrt{\frac{\sum_{i=1}^{n}(Y_i-\hat{y}_i)^2}{(n-2)\sum_{i=1}^{n}(x_i-\bar{x})^2}}$$

β_1 的（1－α）×100%的置信区间为

$$[\beta_0-t_\alpha S_b,\beta_0+t_\alpha S_b]$$

3. 利用回归方程进行预测

利用已通过检验的回归方程，就可以用来预测，即确定自变量的某一个 x 值时求出相应的因变量 Y 的估计值，其中又可以分为点预测和区间预测。

（1）点预测。

将自变量值 x_0 代入回归方程式得到的因变量值 $\hat{y}_0$，作为与 x_0 相对应的 y_0 的预测值，就是点预测。

（2）区间预测。

对于与 x_0 相对应的 y_0，$\hat{y}_0$ 与 y_0 之间总存在一定的抽样误差。在回归模型的假设条件下，有

$$(\hat{y}_0 - y_0) \sim N\left[0, \sigma^2(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_i (x_i - \bar{x})^2}\right]$$

因此，y_0 的概率为 $1-\alpha$ 的预测区间为

$$\hat{y}_0 \pm t_{\frac{\alpha}{2}} \sigma \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_i (x_i - \bar{x})^2}}$$

在实际应用时，一般常采用以下的预测区间

当 α=0.05 时，y_0 的 95%的预测区间为 $\hat{y}_0 \pm 2S_y$。

当 α=0.01 时，y_0 的 95%的预测区间为 $\hat{y}_0 \pm 3S_y$。

式中：$S_y = \sqrt{\frac{\sum_{i=1}^{n} (Y_i - \hat{y}_i)^2}{n-2}}$。

5.2.2 多元线性回归分析

1. 多元线性回归模型

多元线性回归模型是指有多个自变量的线性回归模型，用于揭示因变量与其他多个自变量之间的线性关系，其数学模型为

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

多元线性回归分析就是求得以下的回归方程，并进行相应的检验。

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

与一元线性回归分析一样，可以采用最小二乘方法及其他的优化方法求得多元线性回归方程式中的各个回归系数。

一般地，当（$x_1, x_2, \cdots, x_p, y$）的试验数据为（$x_{i1}, x_{i2}, \cdots, x_{ip}, y_i$），$i=1,2,\cdots,n$ 时，设

$$y=(y_1, y_2, \cdots, y_n), \quad \beta=(\beta_1, \beta_2, \cdots, \beta_p)^{\mathrm{T}},$$

$$\varepsilon=(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p)^{\mathrm{T}}, \quad X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{ni} & \cdots & x_{np} \end{bmatrix}$$

则有

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad \hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y$$

2. 回归方程显著性检验

仍然利用偏差平方和分解公式

$$\sum_{i=1}^{n}(Y_i-\overline{Y})^2=\sum_{i=1}^{n}(Y_i-\hat{y}_i)^2+\sum_{i=1}^{n}(\hat{y}_i-\overline{Y})^2$$

即：$S_T^2=S_R^2+S_E^2$。

回归方程显著性检验，是关于 y 与所有变量 x_i 的线性关系检验，用假设表示为

$$H_0:\ \beta_1=\beta_2=\cdots=\beta_p=0$$

在 H_0 为真时，表明随机变量 y 与 $x_1,x_2,\cdots,x_p$ 之间的线性回归模型不合适，此时的统计量为

$$F=\frac{S_R^2/p}{S_E^2/(n-p-1)}$$

对于给定的数据，计算 F 值，再由给定的显著性水平，查 F 分布表，得临界值 $F_{1-\alpha}(p,n-p-1)$，当 $F>F_{1-\alpha}(p,n-p-1)$ 时，拒绝原假设，即回归方程是显著的。

3. 回归系数显著性检验

对回归系数的线性显著性检验，是关于 y 与某个变量 x_i 的线性关系的检验，用假设表示为 H_0：$\beta_i=0$。当 i=1,2,$\cdots$,p 时，分别关于 y 对 p 个变量进行检验。若接受原假设，则 y 关于 x_i 线性关系不显著，否则显著，此时统计量为

$$T_i=\frac{\hat{\beta}_i}{\sqrt{c_{ii}}\hat{\sigma}}$$

式中：$\hat{\sigma}=\sqrt{\frac{1}{n-p-1}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$；$c_{ii}$ 是矩阵 $(X'X)^{-1}=(c_{ij})$ 的对角线元素；i，j=1,2,$\cdots$,p。

当 $|T_i|>t_{1-\frac{\alpha}{2},n-p-1}$ 时，拒绝原假设，即回归系数是显著的。

4. 拟合检验

定义系数 $R^2=\frac{S_R^2}{S_T^2}=1-\frac{S_E^2}{S_T^2}$ 为相关系数。当 R^2 越接近 1，表明随机因素影响引起的误差越小，回归拟合的效果越好；R^2 越接近 0，表明回归拟合的效果越差。

5. *Y* 的预测区间

当 $Y=\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_px_p+\varepsilon$ 时，由于不确定因素 ε 的影响，只能通过 $\hat{y}$ 对 Y 进行区间估计。

令

$$T=\frac{\dfrac{Y-\hat{y}}{\sigma\sqrt{1+\sum_{j=1}^{p}\sum_{i=1}^{p}c_{ii}x_ix_j}}}{\sqrt{\dfrac{S_E^2}{\sigma^2(n-p-1)}}}=\frac{Y-\hat{y}}{\sqrt{1+\sum_{j=1}^{p}\sum_{i=1}^{p}c_{ii}x_ix_j}\sqrt{\dfrac{S_E^2}{(n-p-1)}}}$$

则，$T \sim t(n-p-1)$，且 Y 的 $1-\alpha$ 预测区间的左右端点为

$$\hat{y}-t_{1-\alpha/2}(n-p-1)\sqrt{1+\sum_{j=1}^{p}\sum_{i=1}^{p}c_{ii}x_i x_j}\sqrt{\frac{S_E^2}{\sigma^2(n-p-1)}}$$

$$\hat{y}+t_{1-\alpha/2}(n-p-1)\sqrt{1+\sum_{j=1}^{p}\sum_{i=1}^{p}c_{ii}x_i x_j}\sqrt{\frac{S_E^2}{\sigma^2(n-p-1)}}$$

5.2.3 非线性回归分析

在实际问题中，变量之间通常不是直线关系，其中的期望函数通常需要根据问题的物理意义或数据点的散布图预先定义，它可以是多项式函数、分式、指数函数以及三角函数等。对于这类回归，可以有两种方法：一是通过变量替换把非线性方程加以线性化，然后按线性回归的方法进行拟合；二是通过适当的优化方法对非线性方程直线进行拟合。

1. 常用的可转化为一元线性回归的模型

常用的非线性转换函数有 y^3、y^2、$y^{1/2}$、ln$y$、$-1/y$、$-1/y^2$ 等。

（1）使 x 上升 y 下降的转换。

对于图 5.1 的情况，可以对 x 进行 x^2、x^3、…的转换，或对 y 进行 ln$y$、$-1/y$…的转换。

（2）使 x 下降 y 上升的转换。

对于图 5.2 的情况，可以对 x 进行 ln$x$、$-1/x$…的转换，或对 y 进行 y^2、y^3…的转换。

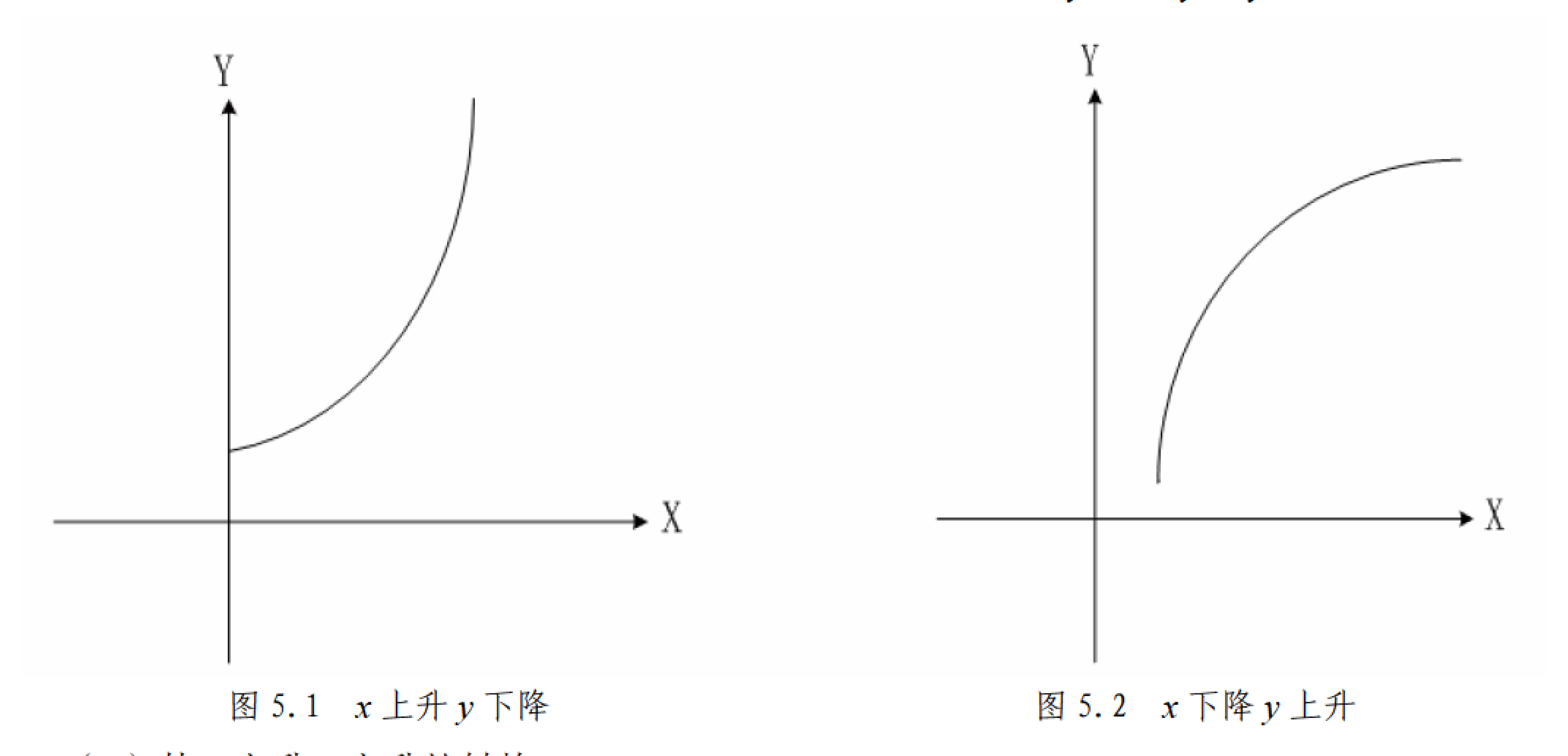


图 5.1　x 上升 y 下降　　　图 5.2　x 下降 y 上升

（3）使 x 上升 y 上升的转换。

对于图 5.3 的情况，可以对 x 进行 x^2、x^3、…的转换，或对 y 进行 y^2、y^3…的转换。

（4）使 x 下降 y 下降的转换。

对于图 5.4 的情况，可以对 x 进行 ln$x$、$-1/x$…的转换，或对 y 进行 ln$y$、$-1/y$…的转换。

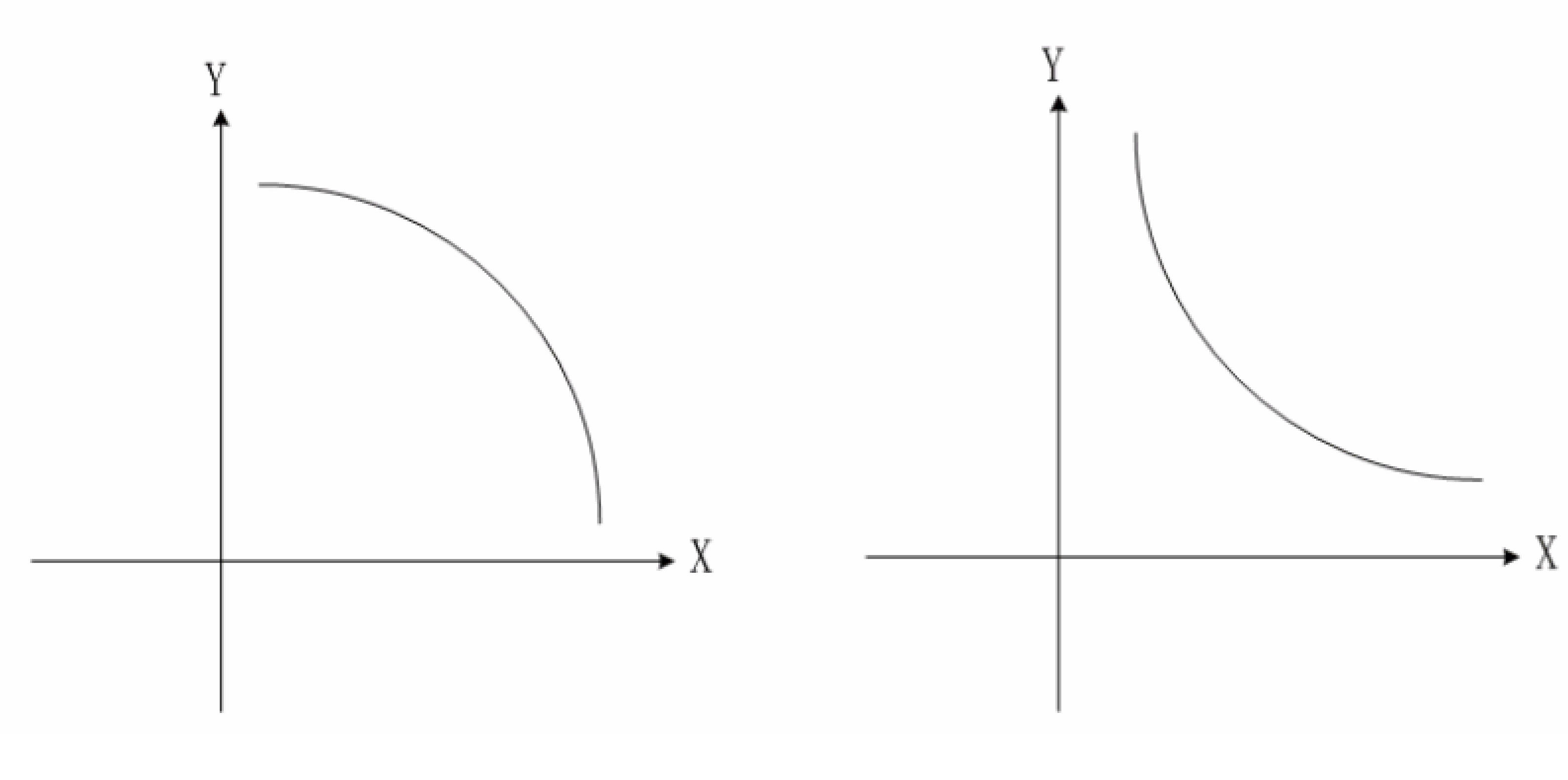


图 5.3　x 上升 y 上升　　　　图 5.4　x 下降 y 下降

以上变换可以用“膨胀规则”来描述，即利用“膨胀规则”来寻求变换以达到变量之间存在线性关系，其步骤如下：

将原始数据曲线图与图 5.5 进行比较，得到 x 与 y 表达阶梯进阶方向，然后根据变量在表达阶梯的位置就可以得到变量变换方式。

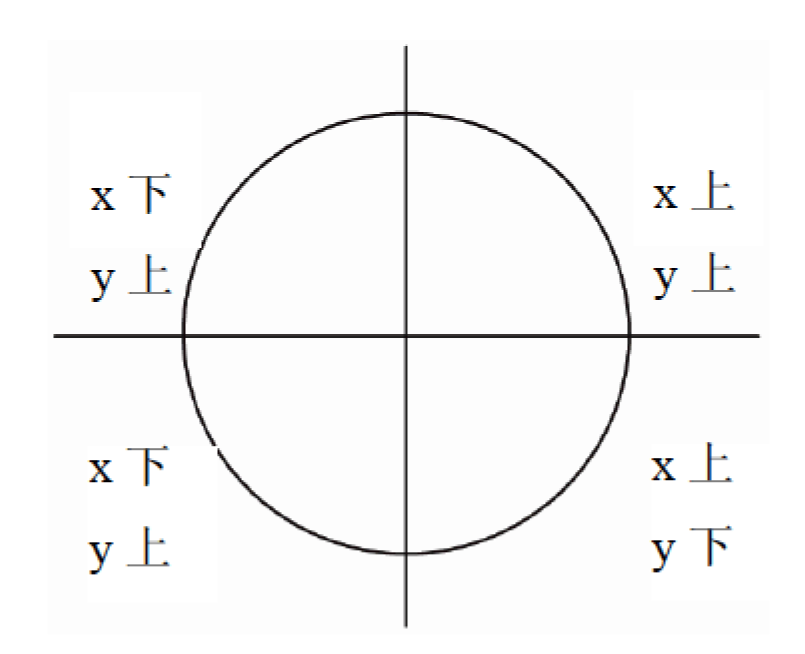


图 5.5　膨胀规则：为实现线性关系，进行启发式的变量变换

表达阶梯包括下列对任何变量 t 的变换集合：

$$t^{-3}t^{-2}t^{-1}t^{-1/2}\ln(t)\sqrt{t}t^{1}t^{2}t^{3}$$

例如将某一原始数据曲线与图 5.5 比较得到曲线的类型为“x 下 y 下”，即表明应该通过将 x、y 从现在的阶梯位置下降一个或多个点来变换变量 x、y。所有未变换变量原始的位置为 t^1。

综上所述，许多曲线都可以通过变换化成直线，于是可以按直线拟合的方法来处理。对变换后的数据进行回归分析，之后将所得的结果再代回原方程。因而，回归分析是对变换后的数据进行，所得结果仅对变换后的数据来说是最佳拟合，当再变换原数据坐标时，所得的回归曲线严格地说并不是最佳拟合，但一般情况下拟合程度还是令人满意的。

2. 一元多项式回归

不是所有的一元非线性函数都能转换成一元线性方程，但任何复杂的一元连续函数都可用多项式近似表达，因此对于那些较难直线化的一元函数，可用下列多项式来拟合。

$$\hat{y}=\beta_0+\beta_1 x+\beta_2 x^2+\cdots+\beta_n x^n$$

同样，通过变量转换或直接多项式拟合的方法可以求出上述方程的各回归系数。

虽然多项式的阶数越高，回归方程与实际数据的拟合程度越高，但阶数越高，回归计算过程中的舍入误差的积累也越大，所以当阶数 n 过高时，回归方程的精确度反而会降低，甚至得不到合理的结果，故一般取 n=3~4。

5.2.4 虚拟及离散变量回归模型

在回归分析的实际应用中，还会遇到虚拟及离散变量为自变量的情况。虚拟变量是指不取实际值的自变量，如性别、国籍、种族、颜色、学位、政治动乱、政府更迭等。如要在回归模型中反映这些因素的影响，可以引入虚拟变量，人为地赋予这些因素一定数据。例如可以引入下列变量至回归方程中。

$$D_i=\begin{cases}1 & \text{第}i\text{个样本来自男性}\\0 & \text{第}i\text{个样本来自女性}\end{cases}$$

当然，也可以赋予其他值，主要取决于实际问题及计算方便性。考虑虚拟变量后的回归方程可以写成下式，其建模方法与一般回归方法相同。

$$Y_i=\beta_0+\beta_1 D_i+\cdots+\beta_i X_i$$

如果虚拟变量有两种以上的取值，则可以使用多个虚拟变量，如学位可以用以下两个虚拟变量：

$$D_{1i}=\begin{cases}1 & \text{学士}\\0 & \text{其他}\end{cases}\qquad D_{2i}=\begin{cases}1 & \text{硕士}\\0 & \text{博士}\end{cases}$$

在地质、医学、经济、生物等科学领域内存在着大量的定性变量，对这些定性变量按一定的方法数量化就可得到离散变量，因此建立离散变量的回归预报方程是一个不可回避的问题。

在建模过程中，称离散变量的不同取值为它的不同水平。考虑离散变量后的回归方程如下所示：

$$y_{k_1,k_2,\cdots,k_m}^{(i)}=\beta_0+\sum_{j=1}^{m}\sum_{k_j=1}^{r_j}\delta_i(j,k_j)\beta_{jk_j}+\cdots+\varepsilon_{k_1,k_2,\cdots,k_m}^{(i)}\qquad i=1,2,\cdots,n$$

式中：$\delta_i(j,k_j)=\begin{cases}1 & \text{第}i\text{次试验第}j\text{个自变量取水平}k_j\text{时}\\0 & \text{其他}\end{cases}$；$y_{k_1,k_2,\cdots,k_m}^{(i)}$ 表示第 i 次试验第 j 个自变量取水平 k_j 时得到的试验结果；rj 为第 j 个自变量水平的数目。

以上模型的解法，可以将其化成方差分析的模型，从而可得各回归系数。具体计算过程参见相关文献或第 4 篇中的例题。

5.2.5 异常点、高杠杆点和强影响观测值

由于各种原因，数据集中各数据的性质并不一样。异常点、高杠杆点和强影响观测值便是其中三种不同性质的数值点。

异常点是指观测到的偏离绝对值很远的一个点，它可以粗略地用标准残留值来评估。如果一个观测点对应的标准残留值的绝对值大于 2，那么就可以认为它是一个异常点。标准残留值的定义如下：

$$\text{residual}_{i,\text{standardized}}=\frac{y_i-\widehat{y_i}}{s_{i,resid}}$$

式中：$s_{i,resid}=s\sqrt{1-h_i}$，$h_i=\frac{1}{n}+\frac{(x_i-\bar{x})^2}{\sum_i(x_i-\bar{x})^2}$，$s$ 为标准差，n 为观测值数量。

高杠杆点可以认为是一个观测值在预测空间中的极限，也即不考虑 y 值的 x 变量的极限，其值可以用杠杆值 h_i 来表示。杠杆值最小可以为 $1/n$，最大为 1。一个拥有大于 $2(m+1)/n$ 和 $3(m+1)/n$（为预测变量的个数）的观测点可以认为是高杠杆点。

强影响观测值是指它的存在将很大程度影响整个曲线的走向，通常强影响观测值同时既有大的残留值又有较高的杠杆。可以通过计算 Cook 距离是否大于 1 确定该点是否具有强影响力。Cook 距离的定义如下：

$$D_i=\frac{(y_i-\widehat{y_i})^2}{(m+1)s^2}\times\frac{h_i}{(1-h_i)^2}$$

如果一个观测值落在分布的第一部分（低于 25%），那么它对整个整体分布只有一点点影响；如果落在分布的中点之后，那么说明该点是具有影响力的。

5.2.6　回归假设检验

前面所介绍的回归方法仅仅依赖于数据和初始回归假设的正确性。在 MATLAB 中可采用统计工具箱中相应的函数来验证回归假设，如检验是否为正态分布的函数有 jbest()、kstest()、lilietest()等函数。如果采用图形化函数如正态概率图（normplot 函数）、回归残差图（rcoplot 函数）则更加直观。

对于正态概率图，如果数据分布是正态的，则大部分的点将落在一条直线上，如果偏离直线表明数据分布不正常。而对于回归残差图，可以有 4 种类型，只有第一种类型（图 5.6（a））才是正常的；（b）图违背了独立性假设；（c）图违背了恒定方差的假设；（d）图违背了零均值的假设。

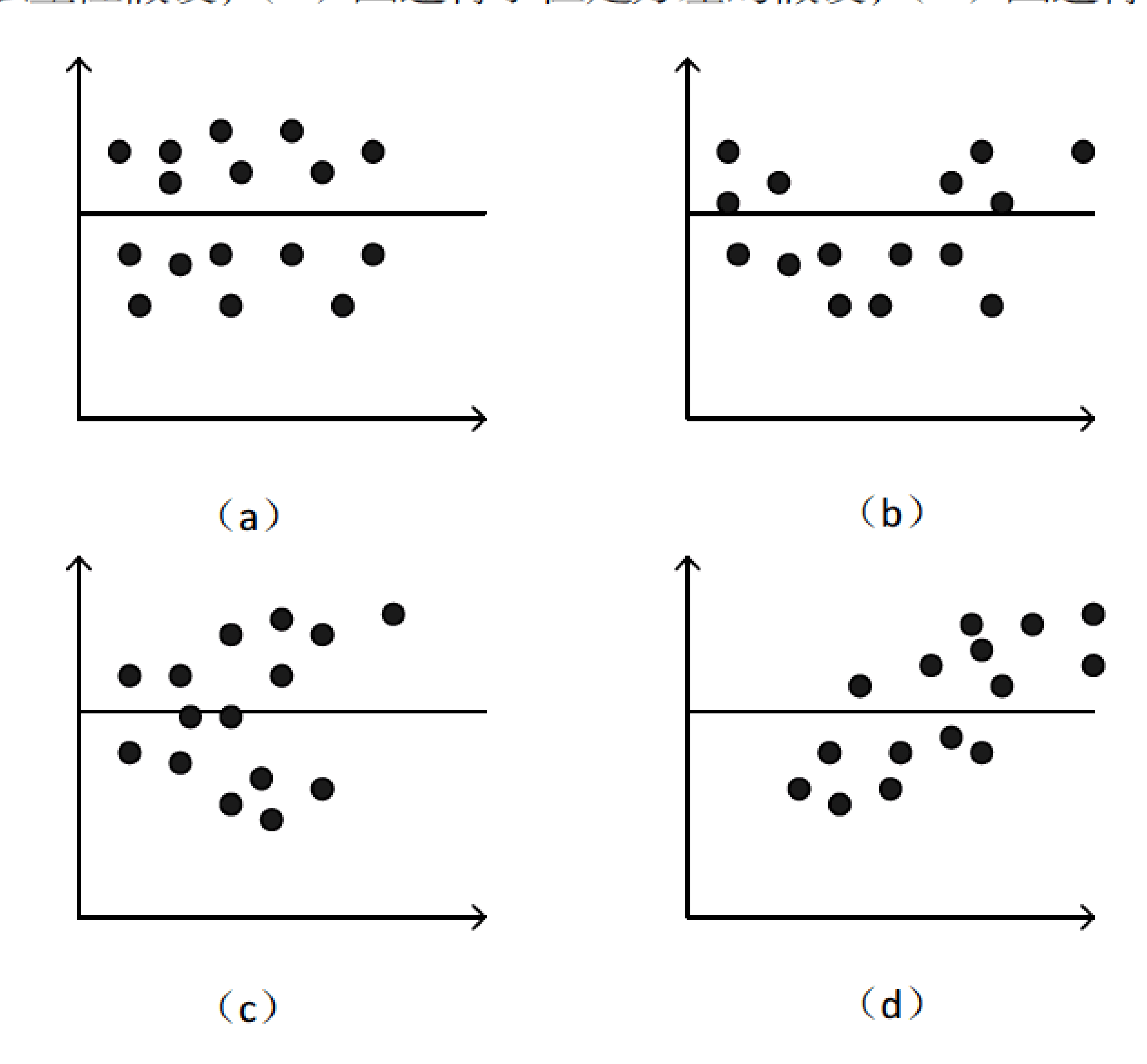


图 5.6　残差与拟合值散点图可能存在的四种模式

5.3 二项逻辑（logistic）回归

多元回归分析在诸多行业和领域的数据分析应用中发挥着极为重要的作用，但是在进行多元回归时，要求因变量是呈正态分布的连续型的随机变量，但在许多问题中，因变量为二值定性变量。例如，在某一药物试验中，动物服药后是生（设其值为 1）还是死（设其值为 0）。显然这时正态线性模型是不合适的。此类问题的解决可借助逻辑回归完成。

5.3.1 二项逻辑回归模型

逻辑回归是根据输入字段值对记录进行分类的一种统计技术。当被解释变量为 0/1 二值品质型变量时，称为二项逻辑回归。

二项逻辑回归虽然不能直接采用一般线性多元回归模型拟合，但仍然可以充分利用线性回归模型建立的理论和思路来拟合。

设因变量 y 为二值定性变量，用 0 和 1 分别表示两个不同的状态，y=1 的概率 p 为研究的对象。自变量 $x_1, x_2, \cdots, x_m$ 可以是定性变量，也可以是定量变量。逻辑回归拟合的回归方程为

$$\ln\frac{p}{1-p} = \beta_0 + \sum_{i=1}^{m}\beta_i x_i$$

式中：m 是自变量个数；p 是在自变量取值为 $X=(x_1, x_2, \cdots, x_m)^{\mathrm{T}}$ 时，因变量 Y 取值为 1 时的概率。$\beta_0, \beta_1, \beta_2, \cdots, \beta_m$ 是待估参数。

逻辑回归方程的另一种形式

$$p = \frac{e^Z}{1+e^Z}$$

其中，$Z = \beta_0 + \sum_{i=1}^{m}\beta_i x_i$ 或 $Z = \ln\frac{p}{1-p}$。

显然 Z 是自变量 X 的线性函数。

今有 c 组实验数据，第 j 组中（j=1,2,$\cdots$,c）试验了 n_j 次，其中 y=1 有 r_j 次，于是概率 p_j 可用 $\hat{p}_j = \frac{r_j}{n_j}$ 来估计，则

$$\hat{Z}_j = \ln\frac{\hat{p}_j}{1-\hat{p}_j} = \beta_0 + \sum_{i=1}^{m}\beta_i x_{ji} \quad (j=1,2,\cdots,c)$$

对上式用加权最小二乘法估计回归系数，即求下式的最小值：

$$\min Q = \sum_{j=1}^{n} W_j (y_j - \hat{y}_j)^2 = \sum_{j=1}^{n} W_j (y_j - (\beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m))^2$$

式中：y_i 和 $\hat{y}_j$ 分别是因变量 y 的第 j 次观察值和预测值；W_j 是给定的第 j 次观察值的权重，一般取观察值误差项方差的倒数：$W_j = \frac{1}{\sigma_j^2}$。但由于一般误差项的方差 σ_j^2 是未知的，所以当 n_j 适当大时，Z_j 的方差可用下式的近似值代替：

$$\sigma^2(Z_j)=\frac{1}{n_j p_j(1-p_j)}$$

可用下式来估计：

$$S^2(Z_j)=\frac{1}{n_j \hat{p}_j(1-\hat{p}_j)}$$

因此，权值数为 $W_j=n_j\hat{p}_j(1-\hat{p}_j)$。

通过微分法可得到 β 的估计值 $\hat{\beta}_j$，记作 b_i。例如在一元逻辑回归中，回归系数为

$$\begin{cases} b_1=\dfrac{\sum W_jX_jZ_j-\dfrac{\sum W_jX_j\sum W_jZ_j}{\sum W_j}}{\sum W_jX_j^2-\dfrac{(\sum W_jX_j)^2}{\sum W_j}} \\ b_0=\dfrac{\sum W_jZ_j-b_1\sum W_jX_j}{\sum W_j} \end{cases}$$

据 $\hat{p}_j=\dfrac{\exp(\beta_0+\beta_1 x)}{1+\exp(\beta_0+\beta_1 x)}$ 画出的曲线呈 S 形，并有两条渐近线 $\hat{p}_j=0$ 和 $\hat{p}_j=1$。

多元逻辑回归方程的系数为

$$\hat{\beta}=(X^TV^{-1}X)^{-1}X^TV^{-1}Z$$

其中

$$X=\begin{bmatrix} 1 & X_{11} & \cdots & X_{1m} \\ 1 & X_{21} & \cdots & X_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{c1} & \cdots & X_{cm} \end{bmatrix}$$

$$\mathrm{V}=\mathrm{diag}[v_1,v_2,\cdots,v_c] \qquad \mathrm{Z}=(z_1,z_2,\cdots,z_c)^{\mathrm{T}}\, Z_j=\ln\frac{\hat{p}_j}{1-\hat{p}_j}$$

V 中的估计值为

$$\hat{v}_j=\frac{1}{n_j\hat{p}_j(1-\hat{p}_j)}$$

如果在 c 组试验结果中，遇到 $r_j=0$ 或 $r_j=n_j$，此时 $\hat{p}_j=0$ 或 $\hat{p}_j=1$，或者遇到 $\hat{p}_j$ 非常接近于 0 或 1，就会出现 $\hat{Z}_j$ 趋于 0 或 1，不再是一个有限值，上述方法就行不通，这时就要对变换和权重进行修正，修正的方法有多种，例如

$$Z_j = \ln\frac{r_j + 0.5}{n_j - r_j + 0.5}$$

$$\hat{v}_j = \ln\frac{(n_j + 1)(n_j + 2)}{n_j(r_j + 1)(n_j - r_j + 1)}$$

5.3.2 显著性检验

Logistic 回归方程的显著性检验包括线性关系检验和回归系数检验两个方面：

1. 回归系数的显著性检验

Logistic 回归方程参数的显著性检验的目的是逐个检验模型中的各个自变量是否与 $\ln\left(\frac{p}{1-p}\right)$ 有显著线性关系，即对解释 $\ln\left(\frac{p}{1-p}\right)$ 是否有重要贡献。检验方法一般采用 Wald 检验。

参数 $\beta_i (i = 1, 2, \cdots, k)$ 的 Wald 统计量定义为 $W = \left[\frac{\hat{\beta}_j}{S_{\hat{\beta}_j}}\right]^2$，其中 $S_{\hat{\beta}_j}$ 为 $\hat{\beta}_j$ 的标准误差，这个单变量 Wald 统计量服从自由度为 1 的 χ^2 分布。

Wald 统计量越大，自变量 $\beta_i (i = 1, 2, \cdots, k)$ 与 $\ln\left(\frac{p}{1-p}\right)$ 之间的关系越显著，应该保留在回归方程中。

2. 线性关系的显著性检验

Logistic 回归方程线性关系的显著性检验的目的是检验全体自变量与 $\ln\left(\frac{p}{1-p}\right)$ 的线性关系是否显著。

Logistic 回归方程显著性的检验一般采用最大似然估计方法。通常将回归模型与截距模型相比较。截距模型没有引入任何自变量，它的似然值最小，是一个“不好”的模型，其定义如下：

$$\ln\left(\frac{p}{1-p}\right) = \beta_0$$

以截距模型作为“基准”，比较当模型中引入自变量后新的模型与数据的拟合水平是否判别显著。差别越大，说明新的模型越有效。其具体步骤如下：

① 定义截距模型，用 L_0 表示截距模型的似然值；

② 构造对数似然化统计量（Likelihood Ratio Test）

$$G^2 = 2\ln\left(\frac{L}{L_0}\right) = (-2\ln L_0) - (-2\ln L)$$

其中：L 为最大似然函数值。$(-2\ln L)$值越大意味着回归模型的似然值越小，模型的拟合程度越差；$(-2\ln L)$值越小则说明回归模型的似然值越大。似然值越接近于 1，模型的拟合程度越好；如果似

然值等于 1，则表示模型完全拟合了观察值。

G^2 近似服从自由度为 k 的 χ^2 分布。

统计量 G^2 越大说明变量全体与 $\ln\left(\frac{p}{1-p}\right)$ 之间的线性关系越显著。

5.3.3　回归方程的拟合优度检验

拟合优度表示回归方程能够解释因变量的变差程度。如果方程可以解释因变量的较大部分变差，则说明拟合优度高，反之则说明拟合优度低。另外，也可以从回归方程的预测准确度来衡量其拟合程度。Logistic 回归方程的拟合优度常用以下两种形式检验。

1. 基于 Cox & Snell R² 统计量的优度检验

Cox & Snell R^2 统计量与一般线性回归的 R^2 有相似之处，也是方程对因变量变差解释程度的反映，其定义为

$$\text{Cox \& Snell } R^2 = 1-\left(\frac{L_0}{L}\right)^{\frac{2}{n}}$$

其中：L_0 是只包含常数项的似然函数值；L 是当前方程的似然函数值；n 为样本量。

Cox & Snell R^2 的取值范围不易确定，解释时有一定困难。

2. 基于 Nagelkerke R² 统计量的优度检验

Nagelkerke R^2 是修正的 Cox & Snell R^2，也反映方程对因变量变差解释的程度，定义为

$$\text{Nagelkerke } R^2 = \frac{\text{Cox \& Snell } R^2}{1-(L_0)^{\frac{2}{n}}}$$

Nagelkerke R^2 的取值在 0 和 1 之间，越接近 1，说明方程的拟合优度越高；值越接近 0，说明方程的拟合优度越低。

5.4　方差分析

在实际应用中，影响事物性质的因素往往有很多，例如产品的销售量与产品的质量、价格、价值、品牌、信誉、偏好等。

任一个因素的改变都有可能影响事物的性质，有的因素影响大些，有的小些。为了使事物的性质稳定，就有必要找出对事物性质有显著影响的那些因素。方差分析就是鉴别各因素效应的一种有效的方法，它主要是指数据的变异、不一致的分析。

5.4.1　单因素试验的方差分析

单因素试验方差分析又称一元方差分析，它是讨论一种因素对试验结果有无显著影响。

设某单因素 A 有 r 种水平 A_1、A_2、⋯、A_r，在各水平下分别做了 n_i（$i=1,2,\cdots,r$）次试验，每种水平下的试验结果服从正态分布，则可以得到表 5.1 所示的数据表。

表 5.1　单因素试验数据表

试验次数	A1	A2	…	Ar
1	x_{11}	x_{21}	…	X_{r1}
2	x_{12}	x_{22}	…	x_{r2}
.	.	.	…	.
.	.	.	…	.
n_j	x_{1n_1}	x_{2n_2}	…	x_{rn_r}

然后根据方差分析的原理，可得到表 5.2 所示的方差分析表。

表 5.2　单因素试验的方差分析表

差异源	SS（方差和）	自由度（df）	均方（MS）	F 检验	显著性
组间（因素 A） 组内（误差 e）	SS_A SS_e	$r-1$ $n-r$	$MS_A=SS_A/(r-1)$ $MS_e=SS_E/(n-r)$	MS_A/MS_e	
总和	SS_T	$n-1$			

表中各物理量的意义及计算方法如下。

$$SS_T=\sum_{i=1}^{r}\sum_{j=1}^{n_i}(x_{ij}-\bar{x})^2 \qquad SS_A=\sum_{i=1}^{r}\sum_{j=1}^{n_i}(\bar{x}_i-\bar{x})^2 \qquad SS_e=\sum_{i=1}^{r}\sum_{j=1}^{n_i}(x_{ij}-\bar{x}_i)^2$$

$$n=\sum_{i=1}^{r}n_i \qquad \bar{x}=\frac{1}{n}\sum_{i=1}^{r}\sum_{j=1}^{n_i}x_{ij}$$

对于给定的显著性水平 α，可查表得到 F 分布的临界值 $F_\alpha(df_A,df_e)$，如计算所得的统计量大于此临界值，说明因素 A 对试验结果有显著影响，否则可以认为 A 对试验结果没有影响。通常，若 $F_A>F_{0.01}(df_A,df_e)$，就称因素 A 对试验结果有非常显著的影响，用两个“*”表示；若 $F_{0.05}(df_A,df_e)<F_A<F_{0.01}(df_A,df_e)$，则称因素 A 对试验结果有显著的影响，用一个“*”表示；若 $F_A<F_{0.05}(df_A,df_e)$，则称因素 A 对试验结果的影响不显著。

应当注意的是，对于单因素多水平的试验，各水平上试验次 n_i 数可以相同，也可以不同，在总的试验次数 n 相同时，n_i 相同时的试验精度更高一些，因此应尽量安排 n_i 相同的单因素多水平试验。

5.4.2　双因素试验的方差分析

根据两因素每种组合水平上的试验次数，可以将双因素试验的方差分析分为无重复试验和重复试验的方差分析。

1. 双因素无重复试验的方差分析

设在某试验中，有两个因素 A 和 B 在变化，A 有 r 种水平 A_1、A_2、…、A_r，B 有 s 种水平 B_1、B_2、…、B_s，在每一种组合水平（A_i,B_j）上做一次试验，试验结果为 x_{ij}（$i=1,2,\cdots r$，$j=1,2,\cdots,s$），所有 x_{ij} 相互独立，得到试验结果如表 5.3 所示。

表 5.3 双因素无重复试验数据表

因 素	B_1	B_2	…	B_s
A_1	x_{11}	x_{12}	.	x_{1s}
A_2	x_{21}	x_{22}	.	x_{2s}
.	.	.	.	.
.	.	.	.	.
A_r	x_{r1}	x_{r2}	.	x_{rs}

对于任一个试验值，其中 r 表示 A 因素对应的水平，s 表示 B 因素对应的水平。显然总试验次数 $n=rs$。

根据方差分析原理，可得到表 5.4 所示的双因素方差分析表。

表 5.4 无重复试验双因素方差分析表

差 异 源	SS（离差）	df（自由度）	MS（均方）	F（统计）	显 著 性
因素 A	SS_A	$r-1$	$MS_A=SS_A/(r-1)$	$F_A=MS_A/MS_e$	
因素 B	SS_B	$s-1$	$MS_B=SS_B/(s-1)$	$F_B=MS_B/MS_e$	
误差	SS_e	$(r-1)(s-1)$	$MS_e=SS_e/((r-1)(s-1))$		
总和	SS_T	$rs-1$			

表中各物理量的意义及计算方法如下。

$$SS_T=\sum_{i=1}^{r}\sum_{j=1}^{s}(x_{ij}-\bar{x})^2 \qquad SS_A=\sum_{j=1}^{s}\sum_{i=1}^{r}(\bar{x}_{i\bullet}-\bar{x})^2 \qquad SS_B=\sum_{i=1}^{r}\sum_{j=1}^{s}(\bar{x}_{\cdot j}-\bar{x})^2$$

$$SS_e=\sum_{i=1}^{r}\sum_{j=1}^{s}(x_{ij}-\bar{x}_{i\cdot}-\bar{x}_{\cdot j}+\bar{x})^2 \qquad \bar{x}=\frac{1}{rs}\sum_{i=1}^{r}\sum_{j=1}^{s}x_{ij} \qquad \bar{x}_{i\bullet}=\frac{1}{s}\sum_{j=1}^{s}x_{ij} \qquad \bar{x}_{\bullet j}=\frac{1}{r}\sum_{i=1}^{r}x_{ij}$$

式中：$\bar{x}$ 为所有试验值的算术平均值，称为总平均；$\bar{x}_{i\bullet}$ 为 A_i 水平时所有试验的算术平均值；$\bar{x}_{\bullet j}$ 为 B_j 水平时所有试验值的算术平均值。

其中：F_A 服从自由度为（df_A，df_e）的 F 分布，对于给定的显著性水平 α，若 $F_A>F_\alpha(df_A,df_e)$，则认为因素 A 对试验结果有显著影响，否则无显著影响；F_B 服从自由度为（df_B，df_e）的 F 分布，若 $F_B>F_\alpha(df_B,df_e)$，则认为因素 B 对试验结果有显著影响。

2. 双因素重复试验的方差分析

在以上方差分析中，是假设两因素相互独立的。但是，在双因素试验中，有时还存在着两因素对试验结果的联合影响，这种联合影响称作交互作用。例如，若因素 A 的数值和水平发生变化时，试验指标随因素 B 的变化规律也发生变化；反之，若因素 B 的数值或水平发生变化，试验指标随因素 A 的变化规律也发生变化，则称因素 A、B 间有交互作用，记为 A×B。如果要检验交互作用对试验指标的影响是否显著，则要求在两个因素的每一个组合（A_i,B_j）上至少做两次试验。

设在某项试验中，有 A、B 两个因素在变化，A 有 r 种水平 A_1、A_2、…、A_r，B 有 s 种水平 B_1、B_2、…、B_s，为研究交互作用 A×B 的影响，在每一种组合水平（A_i,B_j）上重复做 C（$C\geqslant2$）次

试验（称为重复性试验），每个试验值记为 x_{ijk}（i=1,2,⋯r，j=1,2,⋯,s，k=1,2,⋯,c），如表 5.5 所示。

表 5.5　双因素无重复试验数据表

因　素	B_1	B_2	…	B_s
A_1	$x_{111},x_{112},\cdots,x_{11c}$	$x_{121},x_{122},\cdots,x_{12c}$	.	$x_{1s1},x_{1s2},\cdots,x_{1sc}$
A_2	$x_{211},x_{212},\cdots,x_{21c}$	$x_{221},x_{222},\cdots,x_{22c}$	.	$x_{2s1},x_{2s2},\cdots,x_{2sc}$
.	.	.	.	.
.	.	.	.	.
A_r	$x_{r11},x_{r12},\cdots,x_{r1c}$	$x_{r21},x_{r22},\cdots,x_{r2c}$	.	$X_{rs1},x_{rs2},\cdots,x_{rsc}$

然后根据方差分析的原理，可得到表 5.6 所示的方差分析表。

表 5.6　有交互作用双因素试验的方差分析表

差 异 源	*SS*（离差）	*df*（自由度）	*MS*（均方）	*F*（统计）	显 著 性
因素 A	SS_A	$r-1$	$MS_A=SS_A/(r-1)$	$F_A=MS_A/MS_e$	
因素 B	SS_B	$s-1$	$MS_B=SS_B/(s-1)$	$F_B=MS_B/MS_e$	
交互作用	$SS_{A\times B}$	$(r-1)(s-1)$	$MS_{A\times B}=SS_{A\times B}/((r-1)(s-1))$	$F_{A\times B}=MS_{A\times B}/MS_e$	
误差	SS_e	$rs(c-1)$			
总和	SS_T	$rs-1$			

表中各物理量的意义及计算方法如下。

$$SS_T=\sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{c}(x_{ijk}-\bar{x})^2 \qquad SS_A=sc\sum_{i=1}^{r}(\bar{x}_{i\bullet\bullet}-\bar{x})^2 \qquad SS_B=rc\sum_{j=1}^{s}(\bar{x}_{\bullet j\bullet}-\bar{x})^2$$

$$SS_{A\times B}=c\sum_{i=1}^{r}\sum_{j=1}^{s}(\bar{x}_{ij\bullet}-\bar{x}_{i\bullet\bullet}-\bar{x}_{\bullet j\bullet}+\bar{x})^2 \qquad SS_e=\sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{c}(x_{ijk}-\bar{x}_{ij\bullet})^2$$

$$\bar{x}=\frac{1}{rsc}\sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{c}x_{ijk} \qquad \bar{x}_{ij\bullet}=\frac{1}{c}\sum_{k=1}^{c}x_{ijk},i=1,2,\cdots,r;j=1,2,\cdots,s$$

$$\bar{x}_{\bullet j\bullet}=\frac{1}{rc}\sum_{i=1}^{r}\sum_{k=1}^{c}x_{ijk},j=1,2,\cdots,s$$

其中：F_A服从自由度为（df_A，df_e）的 F 分布，对于给定的显著性水平 α，若 $F_A>F_\alpha(df_A,df_e)$，则认为因素 A 对试验结果有显著影响，否则无显著影响；F_B服从自由度为（df_B，df_e）的 F 分布，若 $F_B>F_\alpha(df_B,df_e)$，则认为因素 B 对试验结果有显著影响；$F_{A\times B}$服从自由度为（$df_{A\times B}$，df_e）的 F 分布，对于给定的显著性水平 α，若 $F_{A\times B}>F_\alpha(df_{A\times B},df_e)$，则认为因素 A 对试验结果有显著影响，否则无显著影响。

5.5　主成分分析

在处理多元样本数据时，会遇到一系列问题，如观测数据多，指标间有可能有相关性等。这样它们提供的整体信息会发生重叠，不易得出简明的规律。例如要分析比较若干地区的经济发展状况，对每一个地区都可以统计出数十项与经济状况有关的指标，这些指标虽然能够较详细地反映一个地区的经济发展水平，但要据此对不同地区的发展状况进行评价、比较、排序，则因指标

太多、主次不明显而过于复杂，也很难做到客观公正。另外，在这些指标中，有些是主要的，有些是次要的，甚至某些指标间还有一定的相关性。可以采用主成分分析法来分析这些问题。

主成分分析就是一种把原来多个指标变量转换为少数几个相互独立的综合指标的统计方法。它是通过全面分析各项指标所携带的信息，从中提出一些潜在的综合性指标（即主成分）。

5.5.1　主成分分析的数字模型

设 $X_1, X_2, \cdots, X_p$ 是原始变量，需要求变量 $Z_1, Z_2, \cdots, Z_m$，满足 $m<p$；Z_i 与 Z_j 不相关，即它们间的相关系数为 0，并且 Z_i 能代表 p 个原始变量 x_i 的大部分变异信息，也即降低了原变量的维数。

对 $X_1, X_2, \cdots, X_p$ 观察了 n 次，得到观察数据矩阵为

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

用数据矩阵 X 的 p 个向量（即 p 个指标向量）$X_1, X_2, \cdots, X_p$ 做线性组合为

$$\begin{cases} Z_1 = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\ Z_2 = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\ \quad\vdots \\ Z_p = a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p \end{cases}$$

简写成：

$$Z_i = a_{1i}X_1 + a_{2i}X_2 + \cdots + a_{pi}X_p, i = 1, 2, \cdots, p$$

当全 X_i 是 n 维向量时，Z_i 也是 n 维向量，这里关键是要求 a_{ij}（i,j=1,2,$\cdots$,p；且 $\sum_{j=1}^{p} a_{ij} = 1$）使 $\mathrm{Var}(Z_i)$ 值达到最大。

解约束条件下的 $\mathrm{Var}(Z_i)$ 方程，由于这个解是 p 维空间的一个单位向量，它代表一个“方向”，它就是常说的主成分方向。

一个主成分不足以代表原来的 p 个变量，因此需要寻找第 2 乃至第 3、第 4 主成分，并且每个主成分不应该再包含另外其他主成分的信息，统计上的描述就是让这两个主成分的协方差为零，几何上就是这两个主成分的方向正交。

5.5.2　主成分计算步骤

设 Z_i 表示第 i 个主成分，i=1,2,$\cdots$,p，设

$$\begin{cases} Z_1 = a_{11}X_1 + a_{21}X_2 + \cdots + a_{p1}X_p \\ Z_2 = a_{12}X_1 + a_{22}X_2 + \cdots + a_{p2}X_p \\ \quad\vdots \\ Z_p = a_{1p}X_1 + a_{2p}X_2 + \cdots + a_{pp}X_p \end{cases}$$

其中：对每一个 i，均有 $\sum_{j=1}^{p} a_{ij} = 1$，且（$a_{11}, a_{12}, \cdots, a_{1p}$）使得 Var（$Z_1$）值达到最大；（$a_{21}, a_{22}, \cdots, a_{2p}$）

不仅垂直于（$a_{11}, a_{12}, \cdots, a_{1p}$），而且 Var（$Z_2$）值达到最大；（$a_{31},a_{32},\cdots,a_{3p}$）不仅垂直于（$a_{11},a_{12},\cdots,a_{1p}$）和（$a_{21},a_{22},\cdots,a_{2p}$），而且 Var（$Z_3$）值达到最大；以此类推，直至可求得全部 p 成分。求解的方法就是求 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 矩阵的特征值。

设求得 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的特征值 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p$，它们所对应的标准化正交特征向量为 $\eta_1,\eta_2,\cdots,\eta_p$，则第 1 主成分、第 2 主成分，…，第 p 主成分为

$$Z_1 = \boldsymbol{X}\eta_1$$

$$Z_2 = \boldsymbol{X}\eta_2$$

$$\cdots \quad \cdots$$

$$Z_p = \boldsymbol{X}\eta_p$$

在求解的过程中，要注意以下几点。

① 主成分分析的结果受量纲的影响，由于各变量的单位可能不一样，如果各自改变量纲，结果会不一样，这是主成分分析的最大问题，回归分析是不存在这种情况的，所以实际中可以先把各变量的数据标准化，然后使用协方差矩阵或相关系数矩阵进行分析。

② 为使方差达到最大的主成分分析，所以不用转轴。

③ 主成分的保留。用相关系数矩阵求主成分时，Kaiser 主张将特征值小于 1 的主成分予以放弃。

④ 在实际研究中，由于主成分的目的是为了降维，减少变量的个数，故一般选取少量的主成分（不超过 5 个或 6 个），只要它们能解释变异的 70%～80%（称累积贡献率）就可以了。

5.5.3 主成分估计

$$设\ Z = \begin{bmatrix} Z_{11} & Z_{12} & \cdots & Z_{1p} \\ Z_{21} & Z_{22} & \cdots & Z_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ Z_{n1} & Z_{n2} & \cdots & Z_{np} \end{bmatrix},\quad Q = (\eta_1, \eta_2, \cdots, \eta_p)_{p \times p}$$

$\boldsymbol{Q}$ 为标准化正交阵，且 $\boldsymbol{Z} = \boldsymbol{XQ}$，引入新参数 $\alpha = Q^T \beta$，则主成分回归方程式为

$$Y = \beta_0 1 + Z\alpha + \varepsilon$$

由于特征值≈0 主成分在 n 次试验中取值的变化很小，它的作用可以并入主成分回归方程式中的常数项。因此如果 $\lambda_{r+1}=\cdots=\lambda_p\approx 0$，可剔除 $Z_{r+1},Z_{r+2},\cdots,Z_p$，只保留 α 的前 r 个分量 $\alpha_1,\alpha_2,\cdots,\alpha_r$，设它的最小二乘估计为 $\hat{\alpha}_1, \hat{\alpha}_2, \cdots, \hat{\alpha}_r$ 然后由关系式 $\boldsymbol{\beta} = \boldsymbol{Q\alpha}$ 即可确定 $\boldsymbol{\beta}$ 的估计，这个步骤称为 $\boldsymbol{\beta}$ 的主成分估计。实际步骤如下。

将 $\boldsymbol{Q}$、$\boldsymbol{\alpha}$ 分块，即 $Q = (Q_1, Q_2)$，$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$，其中 Q_1 为 $p\times r$ 矩阵，α_1 为 r 维向量，从而 α 的主成分估计为 $\hat{\alpha} = (\hat{\alpha}_1, 0)^T$，$\beta$ 的主成分估计为 $\hat{\beta} = Q_1\hat{\alpha}_1$。为了增加计算的稳定性，若存在 $1 \leqslant r \leqslant p$，使 $\lambda_r \geqslant 1 > \lambda_{r+1}$，定义

$$A = \mathrm{diag}(\frac{\lambda_1 - 1 + \theta}{\lambda_1}, \cdots, \frac{\lambda_r - 1 + \theta}{\lambda_r}, \theta\lambda_{r+1}, \cdots, \theta\lambda_p)$$

式中：$\theta \in$（λ_p,1）为平稳常数，从而可求得 $\boldsymbol{\beta}$ 的单参数主成分估计

$$\hat{\beta} = QAQ^T Q_1 \hat{\alpha}_1$$

5.5.4　主成分筛选

在进行主成分分析时，判断某主成分是否能删除，一般的依据是删除的特征向量占总特征向量之和的15%以下。但有时仍需考虑选择的主成分对原始变量的贡献值，此时可用相关系数的平方和来表示。如果选取的主成分为$Z_1,Z_2,\cdots,Z_r$，则它们对原变量x_i贡献值为 $\rho_i=\sum_{j=1}^{r}r^2(Z_j,x_i)$。

在选择主成分时，一定要选择与原变量都有关系的主成分，也即如第1主成分不能代表所有变量，则还需要选择第2主成分，以此类推。

5.6　因子分析

因子分析是一种多元统计分析方法，在解决多变量问题时，具有显著的优点。因子分析主要有以下几个优点。

（1）可用于解决很复杂的问题。因子分析作为一种多变量分析方法，可同时处理许多因素相互影响的复杂体系。

（2）能快速地对大量数据进行处理。借助计算机，使用标准的因子分析程序，可以快速地分析大批量数据。

（3）能研究多种类型的问题。在对原始数据了解甚少甚至对数据的本质一无所知的情况下，仍然可应用因子分析方法。这为研究一些未知体系提供了强有力的工具。

（4）可压缩数据，提高数据质量。通过对数据矩阵进行因子分析，可用最少的因子来表示它们，而基本上不损失数据原来所包含的信息，并且还发掘出某些潜在的规则。

（5）可获得对数据的有意义的解释。通过因子分析可对样品或变量进行分类，能够为体系建立完整的有物理意义的模型，以此来预测新的数据点。

5.6.1　因子分析的一般数学模型

因子分析的基本思想是通过变量的相关系数矩阵内部结构的研究，找出能控制所有变量的少数几个随机变量以描述多个变量间的相关系数，通常这少数几个随机变量是不可观察的，称为因子。然后根据相关性大小把变量分组，使得同组内的变量之间相关性较高，但不同组的变量相关性较低。

设 $X_1,X_2,\cdots,X_p$ 是原始变量，影响 X_i 的因素变量有多个，需要寻找少量的公共影响变量反映 X_i 的共同变化规律，即需要确定公共因子变量 $F_1,F_2,\cdots,F_m$ 及特殊因子 ε，使 $m<p$，且

$$X_i=\sum_{j=1}^{m}a_{ij}F_j+\varepsilon_i$$

式中：a_{ij} 是变量 z_i 在因子 F_j 中的载荷。

因子分析的主要问题是：确定每一变量 X_i 的载荷 a_{ij}；确定能反映 p 个原始变量 X_i 变化规律的公共因子个数 F_j，$j=1,2,\cdots,m$，$m<p$；因子旋转，即对确定的 m 个公共因子 F_j 进行解释；因子得分，即代入一组 X_i 的值时，对应 F_j 的取值。

假定有 p 个变量 $X_1, X_2,\cdots, X_p$，在 n 个样品中对这 p 个变量观察的结果组成了如下的原始数

据矩阵：

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ . & . & \cdots & . \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

通常为了消除变量之间在数量级上或量纲上的不同，在进行因子分析之间都要对变量进行如下公式的标准化处理：

$$z_{ji}=\frac{x_{ji}-\overline{x_i}}{\sigma_i} \qquad j=1,2,\cdots,n$$

式中：$\overline{x}_i$ 和 σ_i 分别是第 i 个变量的平均值和标准差。假定标准化以后的变量是 $z_1,z_2,\cdots,z_p$，则标准化数据矩阵是：

$$Z=\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ z_{21} & z_{22} & \cdots & z_{2p} \\ . & . & \cdots & . \\ z_{n1} & z_{n2} & \cdots & z_{np} \end{bmatrix}$$

标准化的目的是使每一个变量的平均值都为零，方差都为 1。

因子分析的基本假设是 p 个标准化变量 $z_1,z_2,\cdots,z_p$，可以是由 m 个新的标准化变量即公共因子 $F_1,F_2,\cdots,F_m$ 线性的组合，如下式表示：

$$\begin{aligned} Z_1 &= a_{11}F_1+a_{12}F_2+\cdots+a_{1m}F_m \\ Z_2 &= a_{21}F_1+a_{22}F_2+\cdots+a_{2m}F_m \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ Z_p &= a_{1p}F_1+a_{p2}F_2+\cdots+a_{pm}F_m \end{aligned}$$

可以证明：

$$\frac{1}{n-1}\sum_{m=1}^{n} f_{mj}z_{mi}=a_{ij} \text{ 及 } 1=a_{i1}^2+a_{i2}^2+\cdots a_{ik}^2 \quad i=1,2,\cdots,p$$

其中：f_{ij} 为因子在某个样品中的得分值。

在计算因子载荷时，需要变换和旋转因子，但不改变特征间的距离，结果因子保持正交，在数学上这样的变换通过解特征值问题得以实现。最佳载荷因子可由因子旋转的方法获得。因子旋转又区分为正交和非正交因子旋转。因子旋转的目的在于使获取的新坐标系统采用最佳的方式将化学测量数据点进行分组，使因子的载荷的结构简单化。

若 L 表示载荷矩阵，L_{rot} 表示旋转后的载荷矩阵，T 表示变换矩阵，则对于正交旋转，有

$$L_{rot}=LT$$

对非正交旋转，若用 L_{fst} 表示因子结构矩阵，L_{fst} 含有公共因子特征的相关的信息，则

$$L_{fst}=LT$$

最常用的是方差最大正交因子旋转。它是一种以因子载荷的方差达到极大为基础的一种正交

因子旋转方法。通过正交变换后，使其中尽可能多的元素接近于零，而只在少数几个特征上有较大的载荷，从而使载荷矩阵的结构简化，有利于做出有意义的解释。

5.6.2　因子模型中公共因子、因子载荷和变量共同度的统计意义

1. 因子载荷的统计意义

因子载荷 a_{ij} 的统计意义就是第 i 个变量与第 j 个因子的相关系数即表示 X_i 依赖 F_j 的分量，即表示第 i 个变量在第 j 个公共因子上的负荷，它反映了第 i 个变量在第 j 个公共因子上的相对重要性。

2. 变量共同度的统计意义

$$h_i^2 = \sum_{j=1}^{m} a_{ij}^2 \qquad i = 1, 2, \cdots, p$$

称为变量 X_i 的共同度，它是 X_i 方差的主要部分，当共同度越大时，说明公共因子包含 X_i 的变异信息就越多。

3. 公共因子的方差贡献的统计意义

$$\mathrm{S}_j = \sum_{i=1}^{p} a_{ij}^2 \qquad j = 1, 2, \cdots, p$$

称为公共因子 F_j 对 X 的贡献，是衡量公共因子相对重要性的指标。

当 $S_{j1}^2 \geqslant S_{j2}^2 \geqslant \cdots \geqslant S_{jp}^2$ 时，对应的公共因子重要性从大到小的排序是 $F_{j1} \geqslant F_{j2} \geqslant \cdots \geqslant F_{jp}$ 。

5.6.3　因子分析与主成分分析的联系与区别

因子分析可以看作主成分分析的推广，是多元统计中常用的降维方法。因子分析所涉及的计算与主成分分析也很相似，两种方法的出发点都是变量的相关系数矩阵，在损失较少信息的前提下，把多个变量（这些变量之间要求存在较强的相关性，以保证能从原始变量中提取主成分）综合成少数几个综合变量来研究总体各方面信息。因此这两种的适用范围是相同的，而且两种方法的综合指标（要使主成分分析中是主成分，在因子分析中是公共因子）与原始指标的关系都是线性的。

因为两种方法有很多相同之处，尤其是在因子分析中用主成分分析方法求解因子载荷时两者似乎更为一致，以致在不少场合将这两种方法不加区分。其实它们之间有联系，也有很大的差异，主要的区别如下：

① 主成分分析仅仅是一种指标变换，不需要任何概率分布和基本统计模型的假定，主要通过少数综合变量反映原始变量的大部分变异信息；而因子分析要假定原始指标所特定的模型，其中的公共因子与特殊因子要满足一定的条件，如标准化与独立性条件等，主要反映原始变量的共同变化规律。

② 主成分分析是将主成分表示为原观察变量的线性组合，即 $Z_i=\sum_{j=1}^{p}\beta_{ij}X_j \quad i=1,2,\cdots,m$ ，一般有 $m<p$，其实质是实现降维，即减少变量个数且反映原始变量的大部分变异信息；则因子分析则是对原始变量分解成公共因子和特殊因子两部分，即 $X_i=\sum_{j=1}^{m}a_{ij}F_j+\varepsilon_i \quad i=1,2,\cdots,n$ ， $m<n$ 其实质也是降维，即把多个原始变量 X_i 看作因变量，少数变量 F_i 看作是自变量，建立这两者间的关系，从而可通过 F_i 的变化研究变量 X_i 的变化。

③ 主成分的各线性系数 β_{ij} 是唯一确定、正交的。不可以对系数矩阵进行任何的旋转，且系数大小并不代表原变量与主成分的相关程度；而因子模型的系数 a_{ij} 是不唯一的，是可以进行旋转的，且系数 a_{ij} 表明了原变量和公共因子的相关程度，旋转使公共因子比主成分更容易解释。

④ 主成分分析可以通过可观察的原变量 X 直接求得主成分 Y，当 $m=p$ 时具有可逆性；因子分析中的载荷是不可逆的，只能通过可观察的原变量去估计不可观测的公共因子，即公共因子得分的估计值等于因子得分系数矩阵与原观察变量标准化后的矩阵相乘的结果。

5.6.4 Q 型和 R 型因子分析

因子分析的起点是协方差或相关矩阵。对 Q、R 型因子分析，由于研究的目的有别，采用的协方差阵也所差别。

$$C_Q=X^TX \qquad (p\times p\text{维})$$
$$C_R=XX^T \qquad (n\times n\text{维})$$

若采用相关矩阵，则

$$R_Q=(XV_Q)^T(XV_Q)$$
$$R_R=(V_RX)(V_RX)^T$$

这里，

$$V_Q(i,j)=\frac{1}{\frac{1}{p-1}\sqrt{\sum_{j=1}^{p}(x_{ij}-\bar{x})^2}}$$
$$V_R(i,j)=\frac{1}{\frac{1}{n-1}\sqrt{\sum_{i=1}^{n}(x_{ij}-\bar{x})^2}}$$

R 型因子分析用于通过 n 次观察研究 P 个特征间的关系；而 Q 型因子分析则是通过 P 个特征来研究 n 个样本间的关系。这两者虽然输入矩阵不一，但因子分析的算法基本一致。

5.7 基于 MATLAB 的统计分析方法

在MATLAB中，有专门的工具箱可以利用。统计工具箱经过不断的发展、完善，现在的版本已经可以与SPSS、SAS等软件的统计功能相媲美。统计工具箱几乎包括了概率论和数理统计的

所有内容，如概率分布、参数估计、方差分析、假设检验、分布检验、聚类分析、判别分析、因子分析、试验设计、统计过程控制、回归分析等。

例2.18　某车间用一台包装机包装糖果。包得的袋装糖重是一个随机变量，它服从正态分布。当机器正常时，其均值为0.5kg，标准差为0.015kg。某日开工后为检验包装机是否正常，随机地抽取它所包装的糖9袋，称得净重为（kg）：0.497 0.506 0.518 0.524 0.498 0.511 0.520 0.515 0.512。问该日的机器工作是否正常?

解：

该题中数据的总体σ已知，$x \sim N(\mu, 0.015^2)$，μ未知。于是提出假设H_0：$\mu=\mu_0=0.5$和H_1：$\mu\neq0.5$

MATLAB实现如下：

```
>>x=[0.497 0.506 0.518 0.524 0.498 0.511 0.520 0.515 0.512];
>>[h,p,ci]=ztest(x,0.5,0.015)
```

求得h=1，p=0.0248，说明在0.05的水平下，可拒绝原假设，即认为这天包装机工作不正常。

例2.19　某种电子元件的寿命x（以小时计）服从正态分布,μ、σ^2均未知。现得16只元件的寿命如下：159 280 101 212 224 379 179 264 222 362 168 250 149 260 485 170。问是否有理由认为元件的平均寿命大于225（小时）?

解：

按题意需检验H_0：$\mu\leqslant\mu_0=225$　　H_1：$\mu>225$

取$\alpha=0.05$。MATLAB实现如下：

```
>>x=[159 280 101 212 224 379 179 264 222 362 168 250 149 260 485 170];
>>[h,p,ci]=ttest(x,225,0.05,1)
```

求得h=0，p=0.2570，说明在显著水平为0.05的情况下，不能拒绝原假设，即可以认为元件的平均寿命不大于225小时。

例2.20　在平炉上进行一项试验以确定改变操作方法的建议是否会增加钢的得率，试验是在同一平炉上进行的。每炼一炉钢时除操作方法外，其他条件都可能做到相同。先用标准方法炼一炉，然后用建议的新方法炼一炉，以后交换进行。每种方法各炼了10炉，其得率分别为

1°标准方法　78.1 72.4 76.2 74.3 77.4 78.4 76.0 75.6 76.7 77.3

2°新方法　79.1 81.0 77.3 79.1 80.0 79.1 79.1 77.3 80.2 82.1

设这两个样本相互独立且分别来自正态总体$N(\mu_1,\sigma^2)$和$N(\mu_2,\sigma^2)$，μ_1、μ_2和σ^2均未知，问建议的新方法能否提高得率?（取α=0.05）

解：

即需检验假设H_0：$\mu_1-\mu_2\geqslant0$　H_1：$\mu_1-\mu_2<0$

MATLAB 实现如下：

```
>>x=[78.1 72.4 76.2 74.3 77.4 78.4 76.0 75.6 76.7 77.3];
>>y=[79.1 81.0 77.3 79.1 80.0 79.1 79.1 77.3 80.2 82.1];
>> [h,p,ci]=ttest2(x,y,0.05,-1)
```

求得h=1，p=2.2126×10^{-4}。表明在α=0.05的显著水平下，可以拒绝原假设，即认为建议的新操作方法较原方法优。

例2.21　下面列出了84个伊特拉斯坎（Etruscan）族人男子头颅的最大宽度（mm），试检验这些数据是否来自正态总体（取α=0.1）。

```
L=[141 148 132 138 154 142 150 146 155 158 150 140 147 148 144 150 149 145 149
158 143 141 144 144 126 140 144 142 141 140 145 135 147 146 141 136 140 146 142
137 148 154 137 139 143 140 131 143 141 149 148 135 148 152 143 144 141 143 147
146 150 132 142 142 143 153 149 146 149 138 142 149 142 137 134 144 146 147 140
142 140 137 152 145]。
```

解：

MATLAB实现如下：

```
    >>clear,clc
x=[141 148 132 138 154 142 150 146 155 158 150 140 147 148 144 150 149 145 149
158 143 141 144 144 126 140 144 142 141 140 145 135 147 146 141 136 140 146 142
137 148 154 137 139 143 140 131 143 141 149 148 135 148 152 143 144 141 143 147 146
150 132 142 142 143 153 149 146 149 138 142 149 142 137 134 144 146 147 140 142 140
137 152 145];
mm=minmax(x)                                  %求数据中的最小数和最大数
hist(x,8)                                     %画直方图
fi=[length(find(x<135)),length(find(x>=135&x<138)),length(find(x>=138&x<142)),...
length(find(x>=142&x<146)),length(find(x>=146&x<150)),length(find(x>=150&x<154)),...
length(find(x>=154))]                         %各区间上出现的频数
mu=mean(x),sigma=std(x)                       %均值和标准差
fendian=[135,138,142,146,150,154]             %区间的分点
p0=normcdf(fendian,mu,sigma)                  %分点处分布函数的值
p1=diff(p0)                                   %中间各区间的概率
p=[p0(1),p1,1-p0(6)]                          %所有区间的概率
chi=(fi-84*p).^2./(84*p)
chisum=sum(chi)                               %皮尔逊统计量的值
x_a=chi2inv(0.9,4)                            %chi2分布的0.9分位数
```

求得皮尔逊统计量chisum=2.2654，$\chi^2_{0.1}(7-2-1)=\chi^2_{0.1}(4)=7.7794$

故在水平0.1下接受H_0，即认为数据来自正态分布总体。

例2.22　合金的强度y与其中的碳含量x有比较密切的关系，今从生产中收集了一批数据，如表5.7所示。

表 5.7　数据集

x	0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18
y	42.0	41.5	45.0	45.5	45.0	47.5	49.0	55.0	50.0

试对表中的数据进行拟合，再用回归分析对它进行检验。

解：

为了确定拟合函数的形式，先画出数据的分布图：

```
>> x=0.1:0.01:0.18;
>> y=[42,41.5,45.0,45.5,45.0,47.5,49.0,55.0,50.0];
>>plot(x,y,'+')
```

得到图5.7，可知y与x大致上为线性关系。

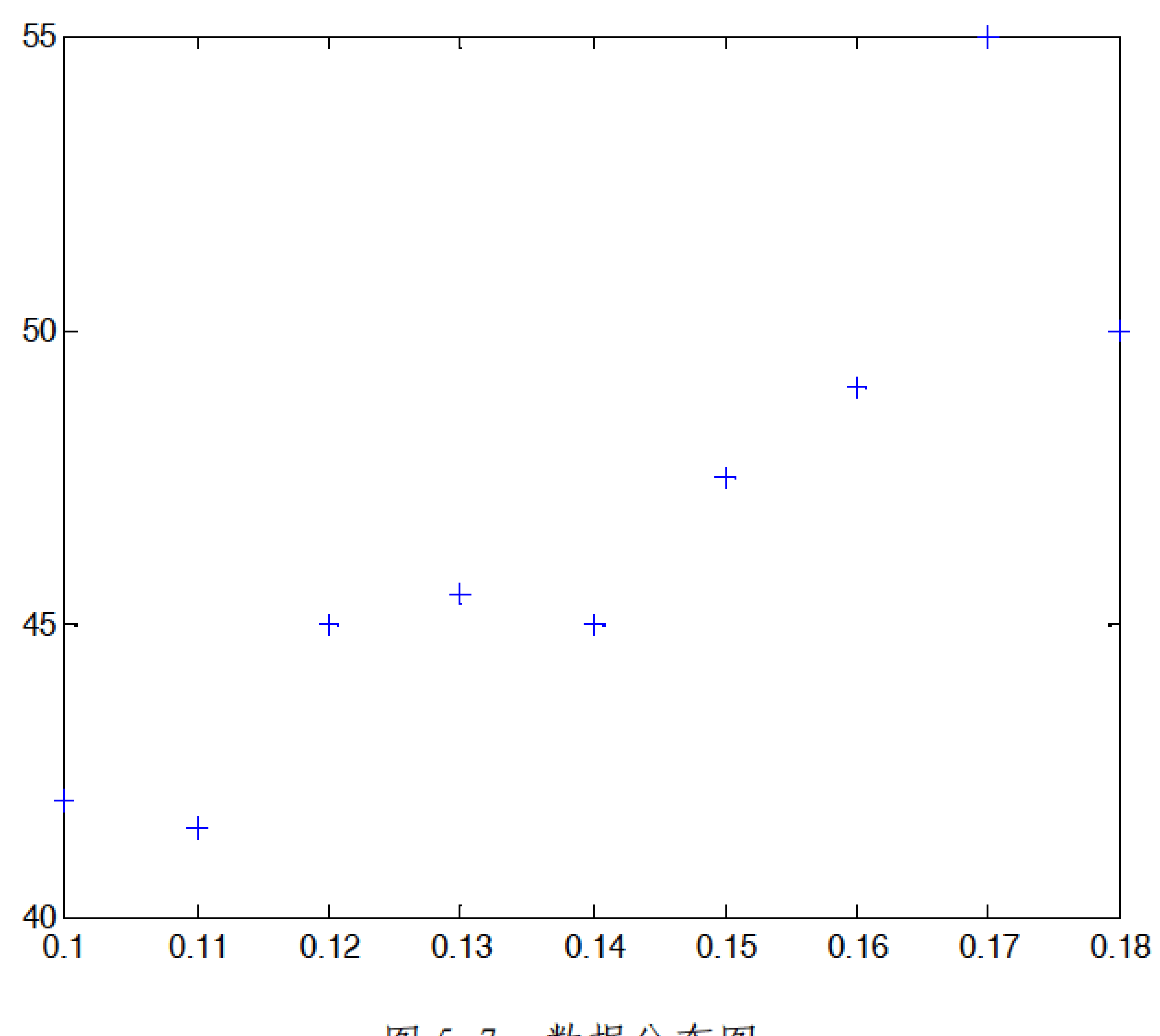


图 5.7　数据分布图

设回归模型为：$y = \beta_0 + \beta_1 x$

MATLAB实现如下：

```
>>clc,clear
x1=[0.1:0.01:0.18]';
y=[42,41.5,45.0,45.5,45.0,47.5,49.0,55.0,50.0]';
x=[ones(9,1),x1];
[b,bint,r,rint,stats]=regress(y,x);
得到b =27.4722 137.5000;
   bint =18.6851 36.2594;
        75.7755 199.2245
```

```
    stats =0.7985 27.7469 0.0012 4.0883
```

即回归系数$\hat{\beta}_0=27.4722$，$\hat{\beta}_1=137.5000$。$\hat{\beta}_0$的置信区间是[18.6851,36.2594]，$\hat{\beta}_1$的置信区间是[75.7755,199.2245]；$R^2=0.7985$，$F=27.7469$，$p=0.0012$，$s^2=4.0883$。可知回归函数式基本符合数据分布。

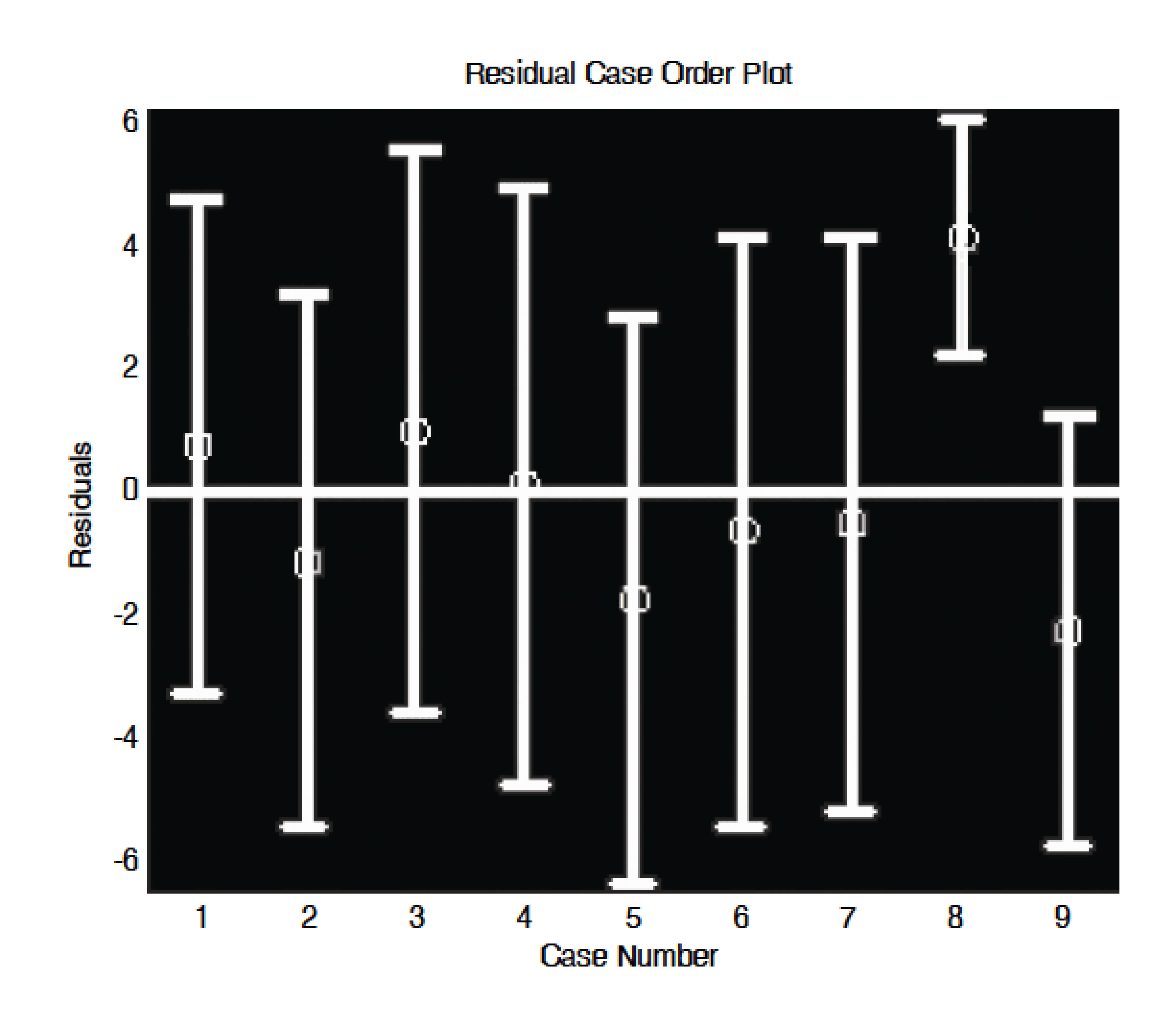


图 5.8　回归数据的残差分布图

再利用回归残差分布图，进一步改进回归方程式。

```
>>rcoplot(r,rint)
```

从图5.8的残差分布图可看出第8个数据的残差置信区间不包含零点，应将此点视为异常点，剔除此点后，再进行回归分析，可得如下结果：

```
  b=30.7820 109.3985
  bint =26.2805 35.2834
        76.9014 141.8955
  stats =0.9188 67.8534 0.0002 0.8797
```

此结果更符合实际情况，应该用修改后的这个结果。

例2.23　某厂生产的一种电器的销售量y与竞争对手的价格x_1和本厂的价格x_2有关。表5.8是该商品在10个城市的销售记录。试根据这些数据建立y与x_1和x_2的关系式，并对得到的模型和系数进行检验。

表 5.8　某电器的销售量数据

x_1	120	140	190	130	155	175	125	145	180	150
x_2	100	110	90	150	210	150	250	270	300	250
y	102	100	120	77	46	93	26	69	65	85

解：

分别画出y关于x_1和y关于x_2的散点图（图5.9），可以看出y与x_2有较明显的线性关系，而y与


x_1之间的关系则难以确定，可以作几种尝试，然后用统计分析决定优劣。

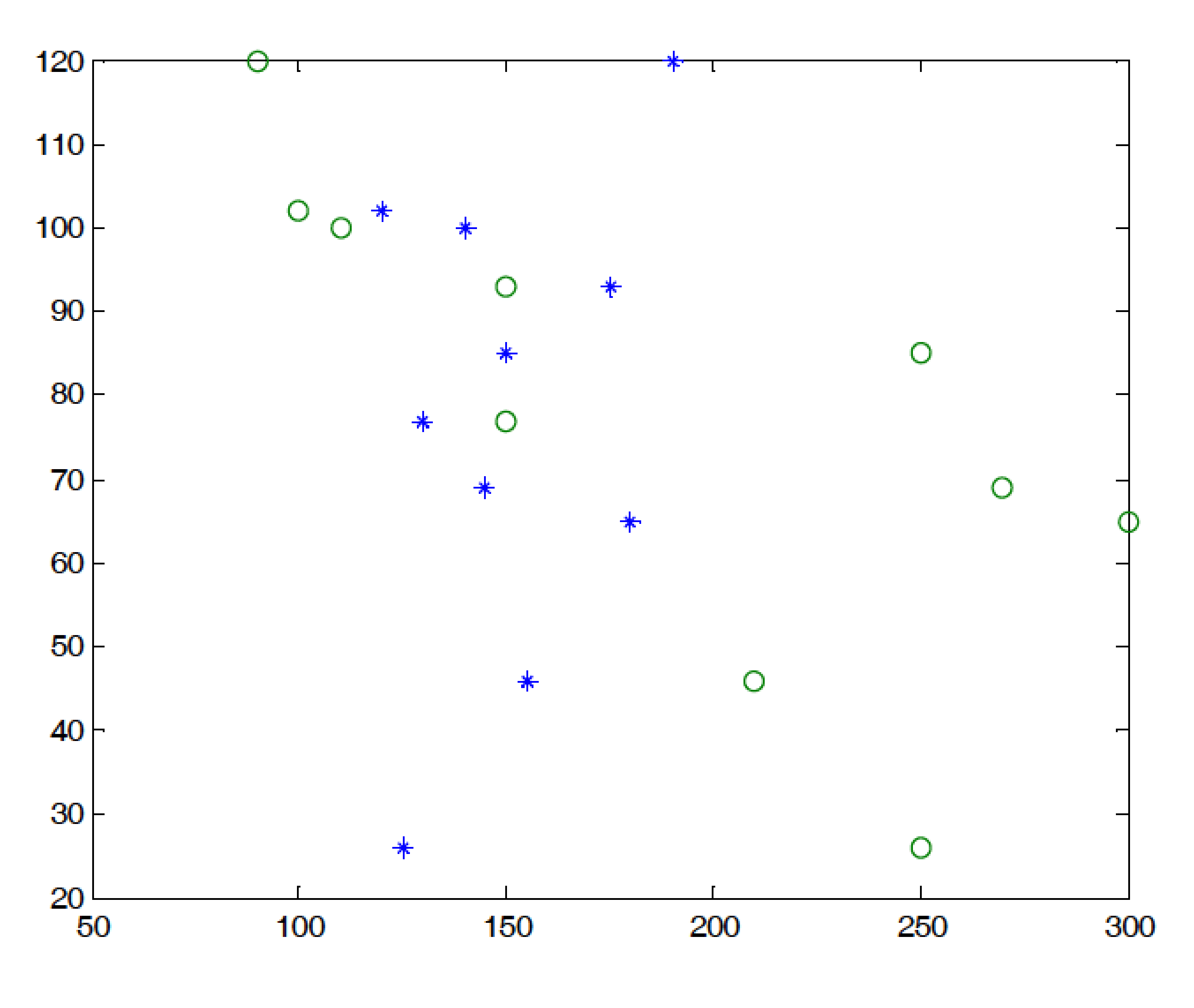


图 5.9 数据分布图

设回归模型为：$y=\beta_0+\beta_1x_1+\beta_2x_2$

编程计算如下。

```
>> x1=[120 140 190 130 155 175 125 145 180 150]';
   x2=[100 110 90 150 210 150 250 270 300 250]';
   y=[102 100 120 77 46 93 26 69 65 85]';x=[ones(10,1),x1,x2];
   [b,bint,r,rint,stats]=regress(y,x);
```

得到如下结果：

```
b=66.5176 0.4139 -0.2698
bint =-32.5060 165.5411
     -0.2018 1.0296
     -0.4611 -0.0785
stats =0.6527 6.5786 0.0247 351.0445
```

可以看出结果不是太好：p=0.0247，取α=0.05时回归模型可用，但取α=0.01则模型不能用；R^2=0.6527较小；$\hat{\beta}_0$、$\hat{\beta}_1$的置信区间包含了零点。

为了得到更好的回归方程式，选用多项式以下回归方法。

设回归模型为

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_{11}x_1^2+\beta_{22}x_2^2$$

在MATLAB工作空间输入：

```
>> x=[x1 x2];rstool(x,y,'purequadratic')
```

得到图5.10所示的交互图。

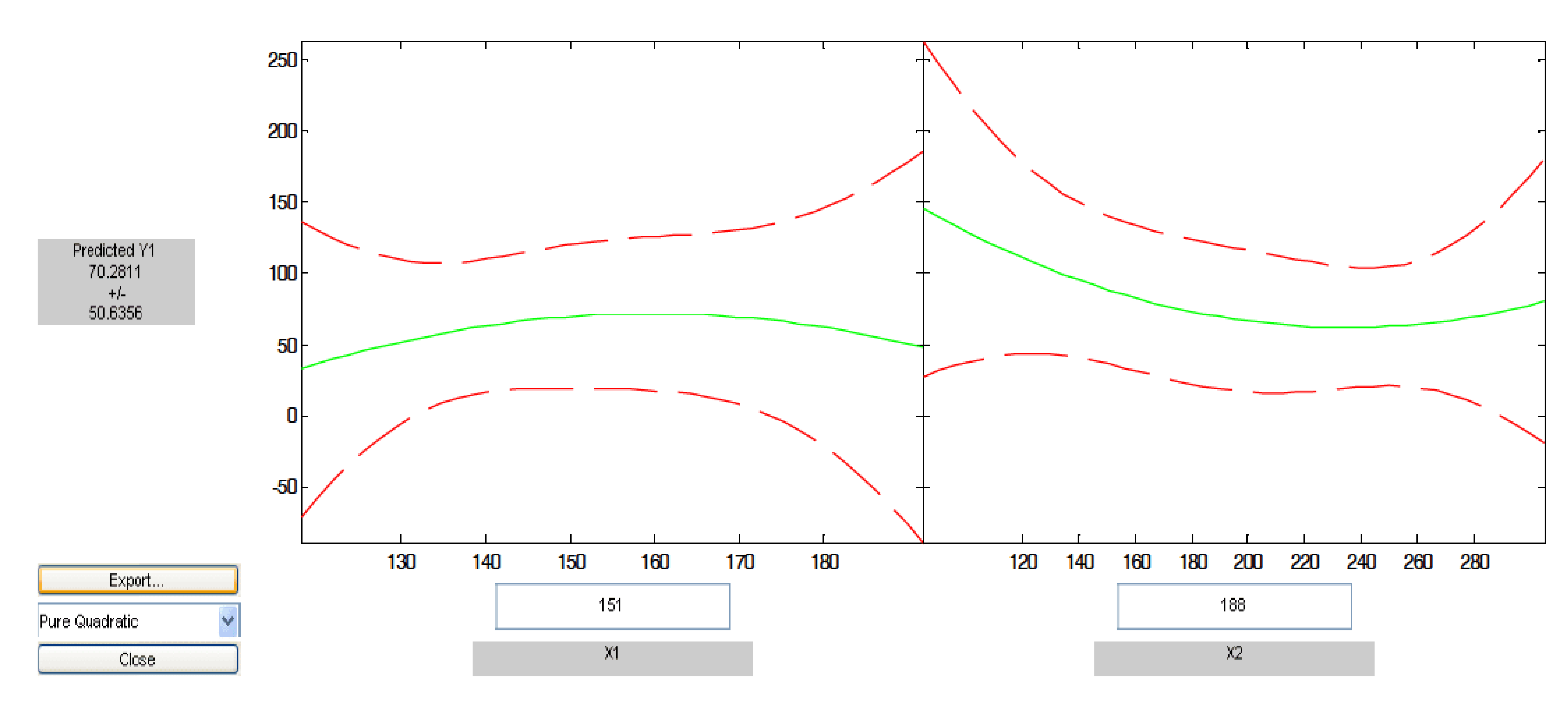


图 5.10 多项式回归交互图

该图的左边是x_1（=151）固定时的曲线$y(x_1)$及其置信区间，右边是x_2（=188）固定时的曲线$y(x_2)$ 及其置信区间。用鼠标移动图中的十字线，或在图下方窗口内输入数值，可改变x_1、x_2。图左边给出y的预测值及其置信区间，用这种画面可以回答诸如“若某市本厂产品售价160（元），竞争对手售价170（元），预测该市的销售量”等问题。

图的左下方有两个下拉式菜单，一个菜单Export用以向MATLAB工作区传送数据，包括beta(回归系数)、rmse（剩余标准差）、residuals（残差）。可得到本题的回归系数和剩余标准差为

```
beta =-312.5871 7.2701 -1.7337 -0.0228 0.0037
rmse =16.6436
```

另一个菜单model用于在以下四个不同的多项式模型中选择，可以通过比较它们的剩余标准差，最终确定回归方程式。

linear(线性）：$y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m$

purequadratic(纯二次): $y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{j=1}^{m}\beta_{jj}x_j^2$

interaction(交叉)： $y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{1\le j\ne k\le m}^{m}\beta_{jk}x_jx_k$

quadratic(完全二次): $y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{1\le j,\ k\le m}^{m}\beta_{jk}x_jx_k$

在本例中最后选择的回归方程式为纯二次多项式，即

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_{11}x_1^2+\beta_{22}x_2^2$$

例 2.24 在研究化学动力学反应过程中，建立了一个反应速度和反应物含量的数学模型，形式为

$$y=\frac{\beta_4x_2-\dfrac{x_3}{\beta_5}}{1+\beta_1x_1+\beta_2x_2+\beta_3x_3}$$

其中：$\beta_1,\cdots,\beta_5$ 是未知的参数；x_1、x_2 和 x_3 是三种反应物（氢、n-戊烷、异构戊烷）的含量；y 是反应速度。今测得一组数据如表 5.9。试由此确定参数 $\beta_1,\cdots,\beta_5$，并给出其置信区间。$\beta_1,\cdots,\beta_5$ 的参考值为（0.1，0.05，0.02，1，2）。

表5.9　实验结果

序号	反应速度y	氢x_1	n-戊烷x_2	异构戊烷x_3	序号	反应速度y	氢x_1	n-戊烷x_2	异构戊烷x_3
1	8.55	470	300	10	8	4.35	470	190	65
2	3.79	285	80	10	9	13.00	100	300	54
3	4.82	470	300	120	10	8.50	100	300	120
4	0.02	470	80	120	11	0.05	100	80	120
5	2.75	470	80	10	12	11.32	285	300	10
6	14.39	100	190	10	13	3.13	285	190	120
7	2.54	100	80	65					

解：

首先，以回归系数和自变量为输入变量，将要拟合的模型写成函数文件myfun1：

```
function  y=myfun1(beta,x)
y=(beta(4)*x(:,2)-x(:,3)/beta(5))./(1+beta(1)*x(:,1)+beta(2)*x(:,2)+beta(3)*x(:,3));
```

然后，用nlinfit函数计算回归系数，用nlparci函数计算回归系数的置信区间，用nlpredci函数计算预测值及其置信区间，编程如下：

```
>>x0=[ 1 8.55 470 300 10;2 3.79 285 80 10;3 4.82 470 300 120;4 0.02 470 80 120
5 2.75 470 80 10;6 14.39 100 190 10;7 2.54 100 80 65;8 4.35 470 190 65;9 13.00 100 300 54
10 8.50 100 300 120;11 0.05 100 80 120;12 11.32 285 300 10;13 3.13 285 190 120];
x=x0(:,3:5);y=x0(:,2);
beta=[0.1,0.05,0.02,1,2]';                          %回归系数的初值,任意取的
[beta1,r,j]=nlinfit(x,y,@myfun1,beta);
beta2=nlparci(beta1,r,'jacobian',j);
beta3=[beta1,beta2]                                 %回归系数及其置信区间
[y2,delta]=nlpredci(@myfun1,x,beta1,r,'jacobian',j)  %y的预测值及其置信区间
y2±delta
```

也可以用nlintool函数得到一个交互式界面来解此题：

```
>>nlintool(x,y,'myfun1',beta);
```

可得到交互式界面，界面中的左下方的Export可向工作区传送数据，如回归系数、剩余标准差等。

例 2.25　根据拼字游戏每个字母频率与分值值数据集（表 5.10），试求频率与字母间回归关系曲线。

表5.10 拼字游戏频率和点值

字　母	频　率	特 征 值	字　母	频　率	特 征 值
A	9	1	N	6	1
B	2	3	O	8	1
C	2	3	P	2	3
D	4	2	Q	1	10
E	12	1	R	6	1
F	2	4	S	4	1
G	3	2	T	6	1
H	2	4	U	4	1
I	9	1	V	2	4
J	1	8	W	2	4
K	1	5	X	1	8
L	4	1	Y	2	4
M	2	3	Z	1	10

解：

在作回归分析前，一般都先画出变量间的散点图，进而判断变量间存在何种关系。对于本例，作图5.11，可以看出，分值与字母频率间的关系并不是直线关系，而更近似于二次关系。

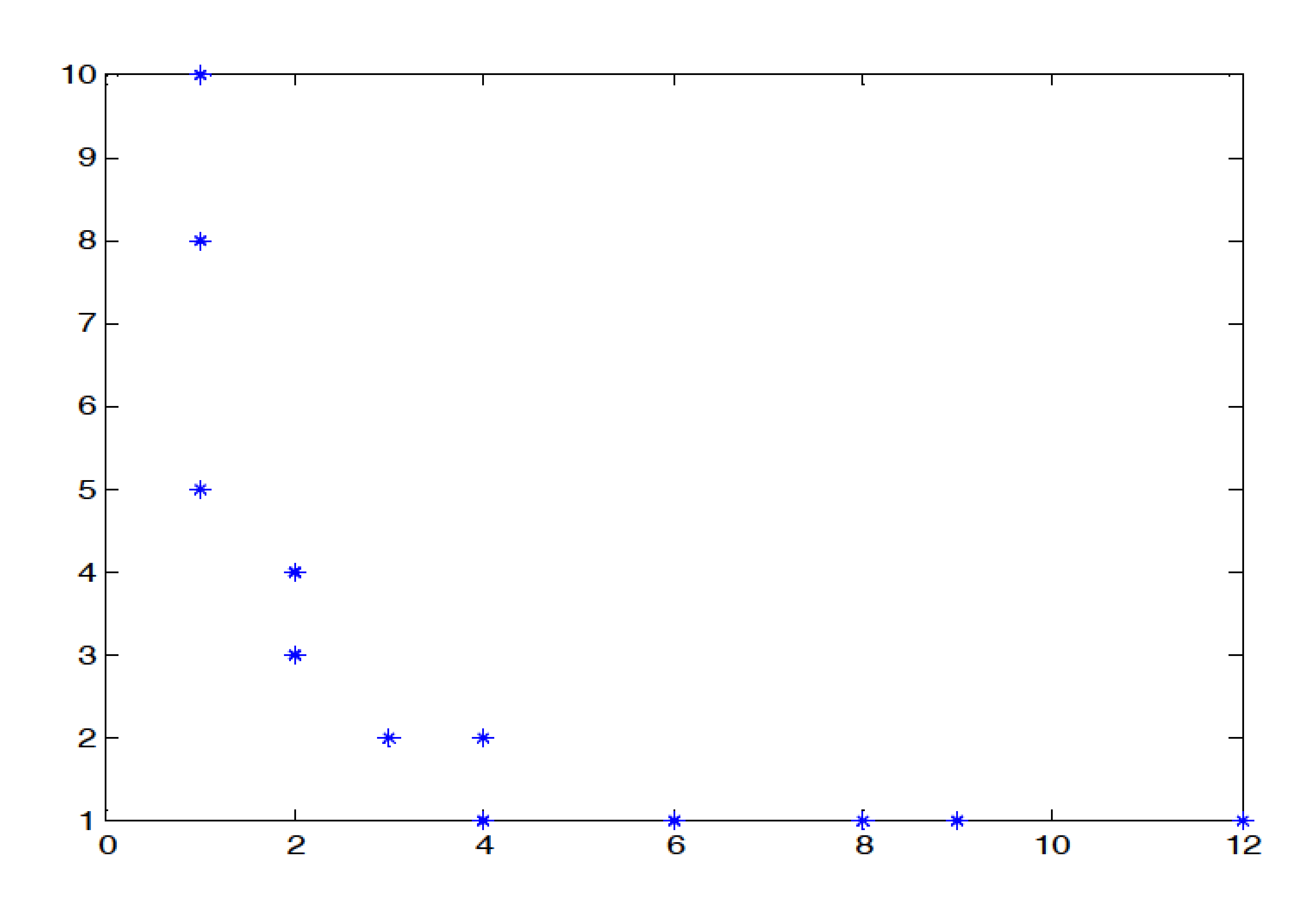


图 5.11 拼字游戏中点相对于频率的散点图

然而，既可以通过转换以实现线性关系，也可以直接进行多式项回归或非线性回归。在此采用第1种方法。

根据“膨胀规则”可以发现图5.10与图5.4中的“x下、y下”的曲线最为相似，因此通过“重新表达阶梯”将现在的阶梯位置上（t^1）下降一个或多个点来变换变量x及y，即使用平方根或自然对数变换可实现线性拟合。图5.12为应用平方根变换后所得到的曲线。可以看出，平方根变换后线性关系仍不明显。所以继续下移，用自然对数变换，得图5.13，可以看出，此时线性关系较

为明显。

最后确定对原始数据进行自然变换后，再进行线性拟合。

```
>> x1=[9 1;2 3;2 3;4 2;12 1;2 4;3 2;2 4;9 1;1 8;1 5;4 1;2 3;6 1;8 1;2 3;1 10;6
1;4 1;6 1;4 1;2 4;2 4;1 8;2 4;1 10];
>> x=[ones(26,1) log(x1(:,1))];
>> [a,b]=regress(log(x1(:,2)),x);
 a=1.9403    -1.0054        %回归系数
```

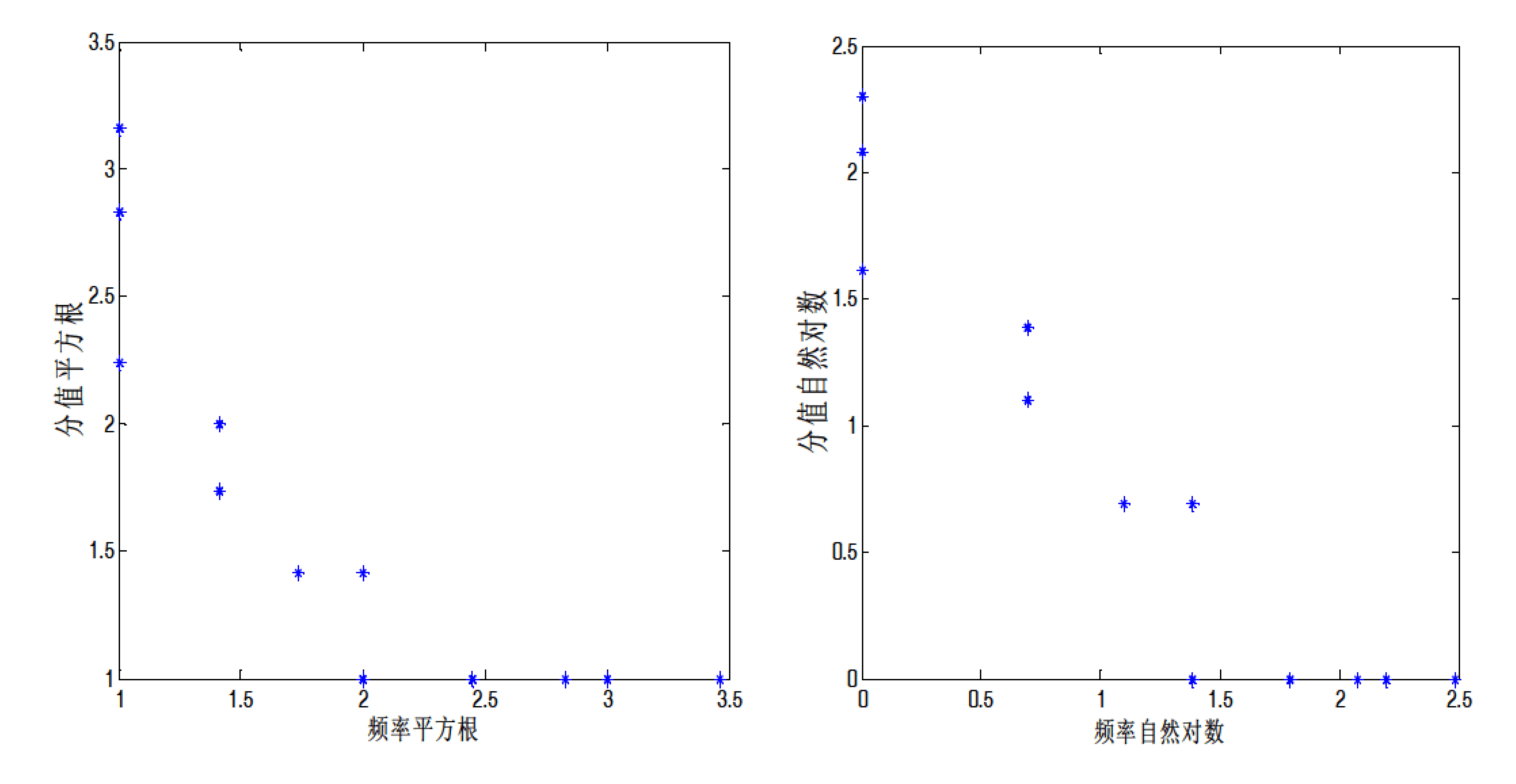


图 5.12　平方根变换后的关系图　　　　图 5.13　自然对数变换后的关系图

例 2.26　在某次住房展销会上，与房地产商签订初步购房意向书的共有 325 名顾客，将 325 名顾客分为 9 组，根据调查，发现在随后的 3 个月的时间内，只有一部分顾客确定购买了房屋。将购买了房屋记为 1，没有购买房屋的顾客记为 0。以顾客的年家庭收入（万元）为自变量，试对表 5.11 中的数据，建立 Logistic 回归模型。

表 5.11　住房展销会历史数据

序　号	年家庭收入 x	签订意向书人数 n_i	实际购房人数 m_i
1	3.5	58	26
2	4.5	52	22
3	5.5	43	20
4	6.5	39	22
5	7.5	28	16
6	8.5	21	13
7	9.5	15	10
8	1.5	25	8
9	2.5	32	13

解：

Logistic 回归方程为

$$p_i = \frac{\exp(\beta_0 + \beta_i x_i)}{1+\exp(\beta_0 + \beta_i x_i)} \qquad i = 1, 2, \cdots, c$$

其中：c 为分组数据的级数，对于本例为 9。

先用非线性回归方法进行回归。

先编写回归方程式：

```
function y=myfun1(beta,x)
y=exp(beta(1)+beta(2)*x)./(1+exp(beta(1)+beta(2)*x));
```

然后，在MATLAB工作空间中输入下列命令：

```
>> x=[3.5 4.5 5.5 6.5 7.5 8.5 9.5 1.5 2.5]; n=[58 52 43 39 28 21 15 25 32];
>> m=[26 22 20 22 16 13 10 8 13]; p=m./n;beta=[0.1,0.05];
>>beta=lsqcurvefit('myfun1',beta,x,p)
beta=-0.9143    0.1648
```

采用先转换成线性关系，再回归：

```
>> x=[ones(9,1),x'];p=log(p./(1-p))';
>> [b,bint,r,rint,stats]=regress(p,x,0.01);
>> b=-0.9187 0.1657
 stats =0.9489  129.8636    0.0000    0.0127
```

两种方法得到的结果基本一致。

从以上的结果可看出，采用一般的方法回归Logistic逻辑回归，效果并不好，需要采用加权偏小二乘法。据此可编程计算得出：

```
>> [b0,b1]=logistic(data)
    b0 =-0.8863  b1 =0.1594
```

即回归方程式为

$$\hat{Z} = -0.886 + 0.16x$$

或

$$\hat{p} = \frac{\exp(-0.886+0.16x)}{1+\exp(-0.886+0.16x)}$$

或写成

$$\hat{p} = \frac{1}{1+\exp(0.886-0.16x)}$$

由回归方程可知，家庭年收入x越高，$\hat{p}$越大，即签订意向后真正购买的概率就越大。例如

对于年收入为9万元的客户，其购买概率为

$$\hat{p} = \frac{1}{1+\exp(0.886-0.16\times 9)} = 0.6309$$

即年收入为9万元的客户签订意向后有63.09%的人会真正买房。

家庭年收入为9万元的客户其签订意向后最终买房与不买房的可能性大小之比为

$$\text{odd}(\text{年收入9万元}) = \frac{\hat{p}}{1-\hat{p}} = \exp(0.886-0.16\times 9) = 1.709$$

说明家庭年收入为9万元的客户其签订意向后最终买房的可能性是不买房的1.709倍。另外，可得如下的关系式：

$$\text{OR}(\text{年收入9万元, 年收入8万元}) = \frac{\exp(-0.886+0.16\times 9)}{\exp(-0.886+0.16\times 8)} = 1.1686$$

所以一个家庭年收入9万元的客户其签订意向后最终买房的可能性是年收入8万元客户的约1.17倍。

例 2.27 表 5.12 给出了一个银行数据的样本，表中第二栏记录了专家对每个银行金融情况的判断，“1”表示金融状况弱，“0”表示金融状况强。表中最后两栏给出了银行金融分析中两个常用比率的值。

表 5.12 银行的金融状况

观 察 点	金融状况（y）	总贷款和租赁/总资产（x_1）	总费用/总资产（x_2）
1	1	0.64	0.13
2	1	1.04	0.10
3	1	0.66	0.11
4	1	0.80	0.09
5	1	0.69	0.11
6	1	0.74	0.14
7	1	0.63	0.12
8	1	0.75	0.12
9	1	0.56	0.16
10	1	0.65	0.12
11	0	0.55	0.10
12	0	0.46	0.08
13	0	0.72	0.08
14	0	0.43	0.08
15	0	0.52	0.07
16	0	0.54	0.08
17	0	0.30	0.09
18	0	0.67	0.07
19	0	0.51	0.09
20	0	0.79	0.13

解：

（1）首先考虑构建一个自变量的简单Logistic回归模型。

首先以银行分类的简单的，以总贷款和租赁与总资产之比（即x_1）作为自变量的Logistic回归模型，将有如下的变量：

因变量： $Y=\begin{cases}1 & \text{如果金融状况窘迫}\\ 0 & \text{其他情况}\end{cases}$

自（或解释）变量：x_1表示“总贷款和租赁与总资产之比”。

因变量和自变量间的关系式为

$$P(Y=1|x_1)=\frac{\exp(\beta_0+\beta_1 x_1)}{1+\exp(\beta_0+\beta_1 x_1)}$$

利用表中数据对模型作最大似然估计，可得：$\widehat{\beta_0}=-6.926, \widehat{\beta_1}=10.99$

即模型为： $P(Y=1|x_1)=\dfrac{\exp(-6.926+10.99x_1)}{1+\exp(-6.926+10.99x_1)}$

很明显：

$$P(Y=0|x_1)=1-P(Y=1|x_1)=\frac{1}{1+\exp(-6.926+10.99x_1)}$$

$$\frac{P(Y=1|x_1)}{P(Y=0|x_1)}=\exp(-6.926+10.99x_1)$$

银行的贷款和租赁与资产之比是0的概率导致财政状况紧张的程度： $\exp(-6.926)=0.001$
这是基本事件的概率。

银行在比率为0.6时在基本事件中财政紧张的概率将按倍数 $\exp(10.99\times 0.6)=730$ 增加，因该银行将陷入财政紧张的概率为0.730。

同样可求得自变量为x_2时的Logistic模型：

$$P(Y=1|x_2)=\frac{\exp(-9.587+94.35x_2)}{1+\exp(-9.587+94.35x_2)}$$

（2）进一步考虑两个变量的Logistic回归模型，其具体表达式为

$$P(Y=1|x1,x2)=\frac{\exp(-14.19+9.173x_1+79.96x_2)}{1+\exp(-14.19+9.173x_1+79.96x_2)}$$

通过对这三个模型的检验，可看出模型三的性能明显要好于另外两个模型，其情况可以利用函数[b,dev,stats]=glmfit()中的stats参数得到。

```
   >> x=[0.64 0.13;1.04 0.10;0.66 0.11;0.80 0.09;0.69 0.11;0.74 0.14;0.63
0.12;0.75 0.12
      0.56 0.16;0.65 0.12;0.55 0.10;0.46 0.08;0.72 0.08;0.43 0.08;0.52
0.07;0.540.08;
      0.30 0.09;0.67 0.07;0.51 0.09;0.79 0.13];
   >> a0=ones(10,1);a1=zeros(10,1);y0=[a0;a1];
   >> theta=glmfit(x(:,1),[y0 ones(20,1)],'binomial','link','logit')  %对x1求模型
```

```
    theta =-6.9258  10.9892
>> theta=glmfit(x(:,2),[y0 ones(20,1)],'binomial','link','logit')  %对x2求模型
    theta =-9.5869  94.3454
>> theta=glmfit(x,[y0 ones(20,1)],'binomial','link','logit')
    theta =-14.1876  9.1732  79.9639
```

对以上回归结果作图，可得图5.14。

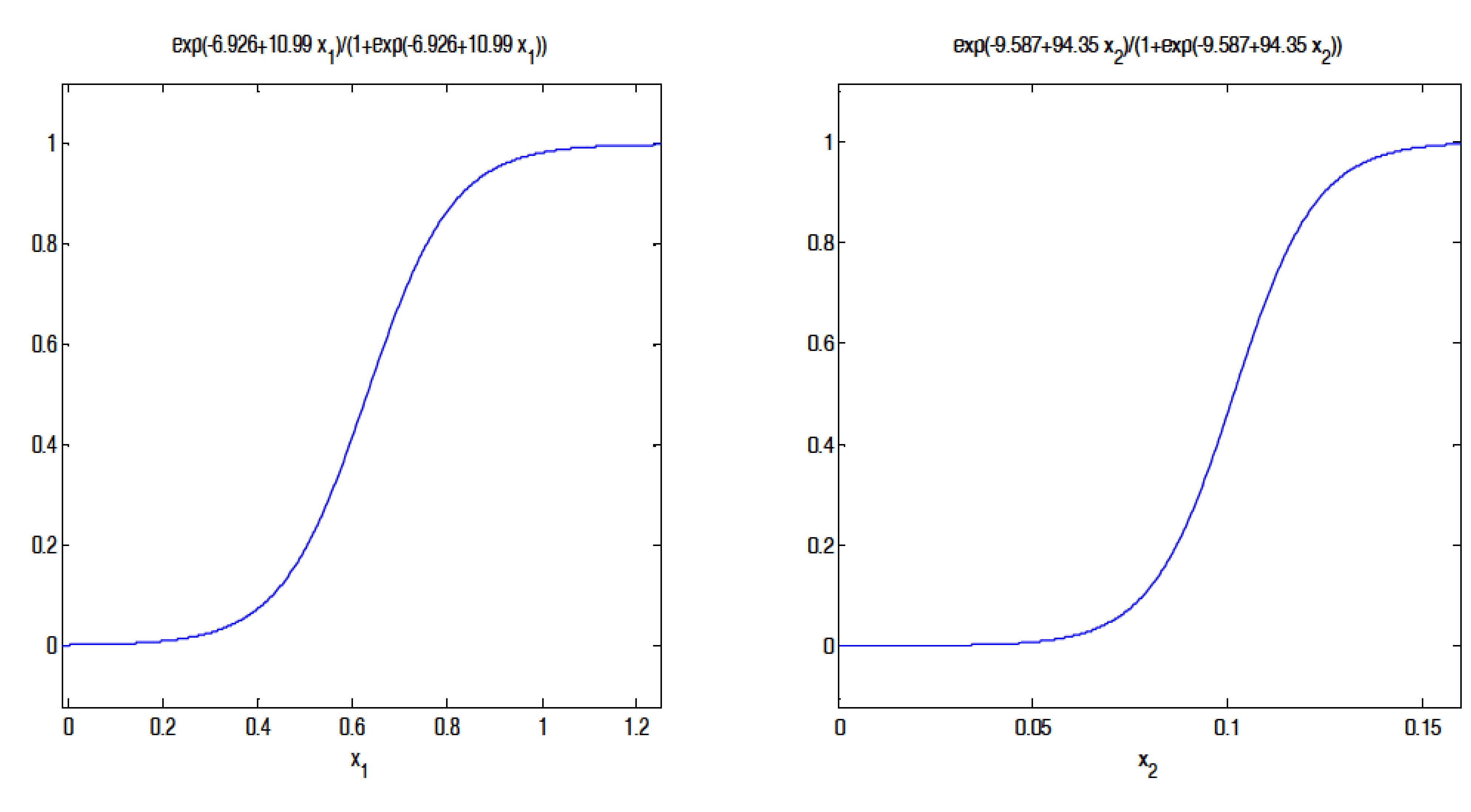


图 5.14　单变量 Logistic 模型图

例 2.28　一般认为，体质指数越大（BMI≥25），表示某人越肥胖，而越肥胖患心血管疾病的概率越大。根据表 5.13 肥胖组患心血管病的体检数据，试建立体质指数与患心血管病概率的逻辑模型。其中表 2.27 中 y 表示是否患心血管疾病，y=1 表示患有，y=0 表示未患有。

表 5.13　肥胖组患心血管的体检数据

体质指数 x	观察值个数	y=1 的观察值个数	y=0 的观察值个数
25	110	68	42
26	93	55	38
27	86	66	20
28	42	32	10
29	28	21	7
30	29	25	4

解：

根据表中的数据，可以进行逻辑回归分析：

```
>> x=[25 26 27 28 29 30]';f=[68 55 66 32 21 25]';t=[110 93 86 42 28 29]';
>> [b,dev] = glmfit(x,[f t],'binomial','logit');
>> b=-6.0324  0.2570     %回归系数
```

于是得到逻辑回归模型为

$$\ln\frac{\hat{p}}{1-\hat{p}}=-6.0323+0.257x$$

从而可知患病概率的拟合值为

$$\hat{p}=\frac{e^{-6.0323+0.257x}}{1+e^{-6.0323+0.257x}}$$

根据BMI和患心血管疾病之间的逻辑关系模型，可以判断出两者间的关系。设体质指数为x_1时，患心血管疾病的概率为p_1，当BMI变化一个单位时，即变为x_1+1时，记患心血管疾病的概率为p_2，则有

$$\ln\frac{\hat{p}_1}{1-\hat{p}_1}=-6.0323+0.257x_1$$

$$\ln\frac{\hat{p}_2}{1-\hat{p}_2}=-6.0323+0.257(x_1+1)$$

$$\ln\frac{\hat{p}_2}{1-\hat{p}_2}-\ln\frac{\hat{p}_1}{1-\hat{p}_1}=\ln\left(\frac{\hat{p}_2/(1-\hat{p}_2)}{\hat{p}_1/(1-\hat{p}_1)}\right)=0.257$$

从而

$$\frac{\hat{p}_2/(1-\hat{p}_2)}{\hat{p}_1/(1-\hat{p}_1)}=\mathrm{e}^{0.257}=1.293$$

这说明

$$\frac{p_2}{1-p_2}\approx 1.293\frac{p_1}{1-p_1}$$

可以看出，BMI对患心血管疾病的影响随着它的增加而增加。

例 2.29　表 5.14 为某公司语音邮箱套餐会员流失的情况，请对此进行逻辑回归分析。

表 5.14　语音套餐会员流失情况统计表

	语音邮箱＝否 x=0	语音邮箱＝是 x=1	合计
流失＝假 y=0	2008	842	2850
流失＝真 y=1	403	80	483
合计	2411	922	3333

解：

根据表中的数据可以得到使用语音邮箱套餐的客户流失的发生比（事件发生的概率与事件不发生的概率之比）

$$p(y=1\,|\,x=1)=\frac{p_1}{1-p_1}=\frac{80}{842}=0.0950$$

未使用语音套餐的客户流失的发生比

$$p(y=1\,|\,x=0)=\frac{p_0}{1-p_0}=\frac{403}{2008}=0.2007$$

从而可得到让步比（是指x=1时因变量发生的发生比除以x=0时因变量发生的发生比）

$$\mathrm{OR}=\frac{\dfrac{p_1}{1-p_1}}{\dfrac{p_0}{1-p_0}}=\frac{0.095}{0.2007}=0.47$$

从以上两个数据可分别计算出逻辑回归的系数

$$b_1=\ln 0.47=-0.7550$$

$$b_0=\ln(0.095/0.47)=-1.5989$$

则拥有语音邮箱套餐的客户或者没有语音邮箱套餐的客户流失的估计量为

$$\hat{p}=\frac{\mathrm{e}^{-1.5989-0.7550x}}{1+\mathrm{e}^{-1.5989-0.7550x}}$$

对于一个拥有此套餐的客户，估计其流失的概率为

$$\hat{p}=\frac{\mathrm{e}^{-1.5989-0.7550\times1}}{1+\mathrm{e}^{-1.5989-0.7550\times1}}=0.0868$$

此概率要小于客户流失的总比例（483/3333 = 14.5%），说明开通语音邮箱套餐有利于减少客户的流失。

对于没有开通语音邮箱套餐的客户，估计其流失的概率为

$$\hat{p}=\frac{\mathrm{e}^{-1.5989}}{1+\mathrm{e}^{-1.5989}}=0.1681$$

此概率比客户流失的总比例稍高一点，说明没有开通语音邮箱套餐对客户流失的影响并不大。

例 2.30　判断客户是否会流失，客服电话数也是一个较好的变量（CSC）。对例 2.29 中数据按其拨打客服电话数进行统计，可以得到表 5.15 所示的数据集。在此 CSC _ 低是指拨打 0 个或 1 个客服电话；CSC _ 中是指拨打 2 个或 3 个客服电话；CSC _ 高是指拨打 4 个或以上的客服电话。试对其进行逻辑回归分析。

表 5.15　语音邮箱套餐会员流失情况统计表

	CSC _ 低	CSC _ 中	CSC _ 高	合　计
流失 = 假 y=0	1664	1057	129	2850
流失 = 真 y=1	214	131	138	483
合计	1878	1188	267	3333

解：

根据题意可知，CSC 是一个三分预测变量。对于这类问题，首先需要用指示变量（虚拟变量）

和参照单元编码法给数据集编码。假定选择 CSC _ 低作为参照单元，则可把指标变量值分配给另外两个变更－CSC _ 中和 CSC _ 高，如表 5.16 所示。

表 5.16 使用参考单元编码的

	CSC _ 中	CSC _ 高
低（0 个或 1 个电话）	0	0
中（2 个或 3 个电话）	1	0
高（≥4 个电话）	0	1

把CSC _ 低作为参考单元，则可计算出让步比：

对于CSC _ 中：

$$\mathrm{OR} = \frac{131\times1664}{214\times1057} = 0.963687 \approx 0.96$$

$$b_1 = \ln 0.96 = -0.0369891$$

对于 CSC _ 高：

$$\mathrm{OR} = \frac{138\times1664}{214\times129} = 8.31819 \approx 8.32$$

$$b_2 = \ln 8.32 = 2.11844$$

因为，对于那些很少拨打电话的客户的流失率为

$$p(y=1\,|\,\text{CSC__ 低}) = \frac{214}{1878} = 0.114$$

从这个值可以求出 b_0：

$$\hat{p} = \frac{1+e^{b_0+b_1(\quad_中)\quad b_2(\mathrm{CSC}_高)}}{1+e^{b_0+b_1(\quad_中)+b_2\quad_高)}} = \frac{1+e^{b_0+b_1(0)+b_2(0)}}{1+e^{b_0+b_1(0)+b_2(0)}} = 0.114$$

$$b_2 = \ln\frac{0.114}{1-0.114} = -2.0505$$

所以，客户流失概率的估计量为

$$\hat{p}(y=1) = \frac{1+e^{b_0+b_1(\quad_中)\quad b_2(\mathrm{CSC}_高)}}{1+e^{b_0+b_1(\quad_中)+b_2(\mathrm{CSC}_高)}} = \frac{1+e^{-2.051-0.0369891(\mathrm{CSC}_中)+2.11844(\mathrm{CSC}_高)}}{1+e^{-2.051-0.0369891(\mathrm{CSC}_中\ +2.11844(\mathrm{CSC}_高}}$$

从而可以计算出以下各种情况下的流失概率：

拨打电话处于中等水平的客户的流失概率：

$$\hat{p}(y=1) = \frac{e^{-2.051-0.0369891(0)+2.11844(0)}}{1+e^{-2.051-0.0369891(1)+2.11844(0)}} = 0.11$$

与很少拨打电话的客户流失概率基本相等，所以可以不考虑 CSC _ 低和 CSC _ 中的客户流失率之间的差异。

经常拨打电话的客户的流失概率：

$$\hat{p}(y=1)=\frac{e^{-2.051-0.0369891(0)+2.11844(1)}}{1+e^{-2.051-0.0369891(0)+2.11844(1)}}=0.5169$$

有一个较高的流失率，比全部样本的客户流失率要高 3 倍，显然，公司要注意这些拨打电话不少于 4 个的客户。

例 2.31　水泥凝固时放出的热量 y 与水泥中 4 种化学成分 x_1、x_2、x_3、x_4 有关，今测得一组数据如表 5.17 所示。试用逐步回归来确定一个回归模型。

表 5.17　实验数据

序　号	x_1	x_2	x_3	x_4	y
1	7	26	6	60	78.5
2	1	29	15	52	74.3
3	11	56	8	20	104.3
4	11	31	8	47	87.6
5	7	52	6	33	95.9
6	11	55	9	22	109.2
7	3	71	17	6	102.7
8	1	31	22	44	72.5
9	2	54	18	22	93.1
10	21	47	4	26	115.9
11	1	40	23	34	83.8
12	11	66	9	12	113.3
13	10	68	8	12	109.4

解：

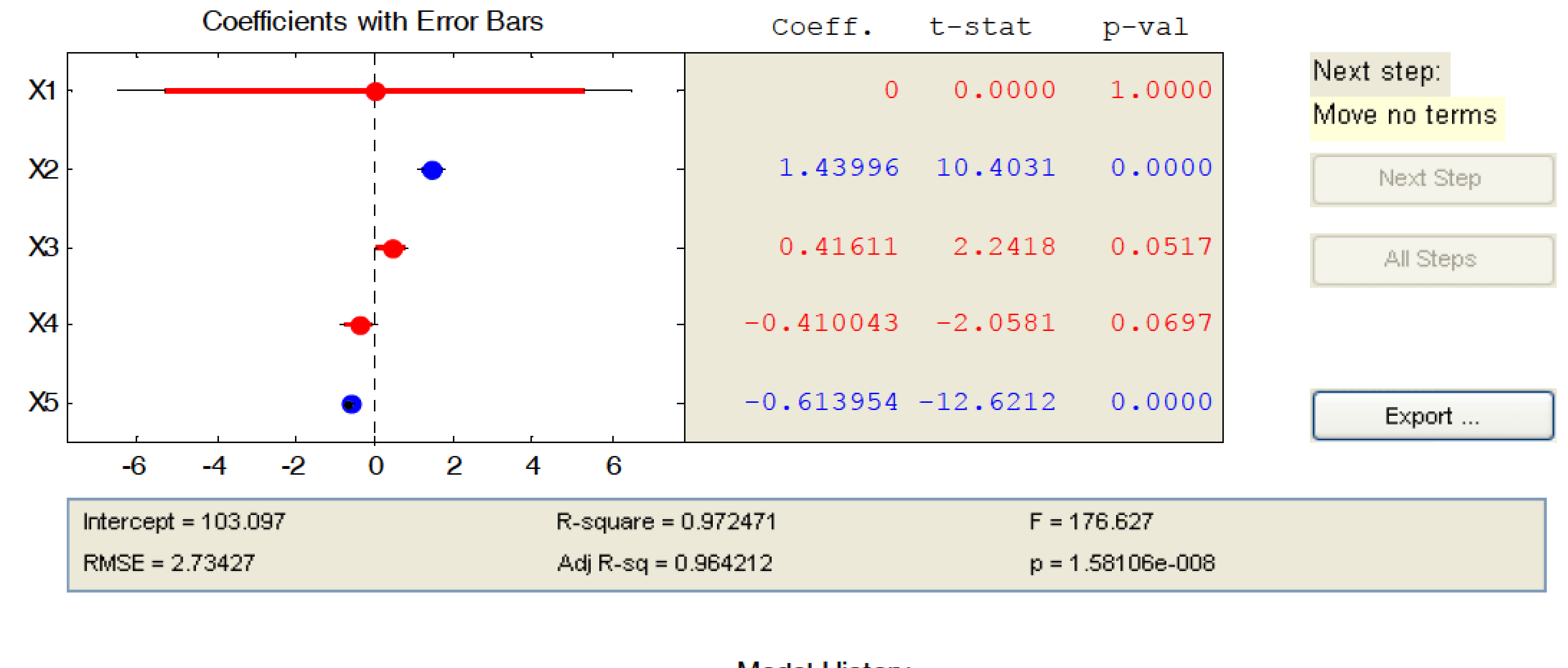
```
>>x0=[7 26 6 60 78.5;1 29 15 52 74.3;11 56 8 20 104.3;11 31 8 47 87.6;
      7 52 6 33 95.9;11 55 9 22 109.2;3 71 17 6 102.7;1 31 22 44 72.5;
      2 54 18 22 93.1;21 47 4 26 115.9;1 40 23 34 83.8;11 66 9 12 113.3;
      10 68 8 12 109.4];
     x=x0(:,1:4);y=x0(:,5);
     x=[ones(13,1),x]      %加入常数项
   >>stepwise(x,y)         %逐步回归函数
```

得到图5.15所示的图形界面。根据界面中的提示，逐步对变量进行移出或移入等操作，最后得到结果（图中显示蓝色的为最后选中的变量）：

```
  beta1=01.440000-0.6140
```

即回归方程为

$$y=1.44x_1-0.614x_4$$



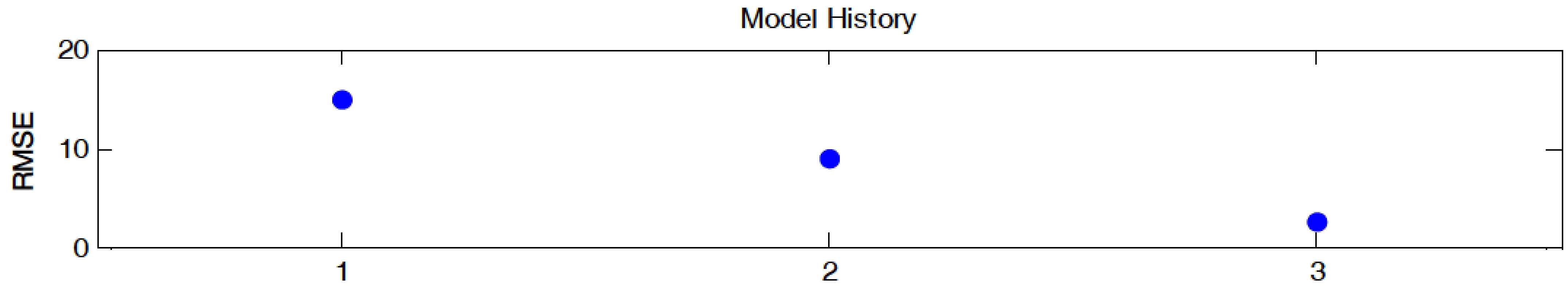


图 5.15 逐步回归交互式画面

例 2.32 为考察 5 名工人的劳动生产率是否相同，记录了每人 4 天的产量，并算出其平均值，得到如表 5.18 所示的结果。请判断他们的生产率有无显著差别。

表 5.18 实验数据表

工人 / 天	A_1	A_2	A_3	A_4	A_5
1	256	254	250	248	236
2	242	330	277	280	252
3	280	290	230	305	220
4	298	295	302	289	252
平均产量	269	292.25	264.75	280.5	240

解：

```
>>x=[256 254 250 248 236;242 330 277 280 252;280 290 230 305 220;298 295 302
    289 252];
>>p=anova1(x); %单因素方差分析
```

得到如下的方差分析表（从方差分析图形中或直接命令得到）：

```
Source    SS      df    MS       F     Prob>F
Columns   6125.7   4   1531.43  2.26   0.1109
  Error   10156.5  15   677.1
  Total   16282.2  19
```

因 $p = 0.1109 > \alpha = 0.05$，故接受 H_0，即 5 名工人的生产率没有显著差异。

例 2.33 用 4 种工艺生产灯泡，从各种工艺制成的灯泡中各抽出了若干个测量其寿命，结果如表 5.19 所示，试推断这几种工艺制成的灯泡寿命是否有显著差异。

表 5.19 实验数据表

工艺 / 序号	A_1	A_2	A_3	A_4
1	1620	1580	1460	1500
2	1670	1600	1540	1550
3	1700	1640	1620	1610
4	1750	1720		1680
5	1800			

解：

```
>>x=[1620 1580 1460 1500;1670 1600 1540 1550;1700 1640 1620 1610;1750 1720 1680
1800];
x=[x(1:4),x(16),x(5:8),x(9:11),x(12:15)];g=[ones(1,5),2*ones(1,4),3*ones(1,3);
4*ones(1,4)];
p=anova1(x,g)
```

得到如下的结果：

```
        Source      SS       df      MS         F       Prob>F
        Groups     62820      3     20940      4.06    0.0331
        Error      61880     12    5156.67
        Total     124700     15
```

因0.01<p=0.0331<0.05，所以几种工艺制成的灯泡寿命有显著差异。

例 2.34 一种火箭使用了四种燃料、三种推进器，进行射程试验，对于每种燃料与每种推进器的组合做一次试验，得到试验数据如表 5.20 所示。请问各种燃料之间及各种推进器之间有无显著差异？

表 5.20 火箭射程试验数据

	B_1	B_2	B_3
A_1	58.2	56.2	65.3
A_2	49.1	54.1	51.6
A_3	60.1	70.9	39.2
A_4	75.8	58.2	48.7

解：

设燃料因素用A表示，它有4个水平，水平效应为α_i，i=1,2,3,4；推进器因素为B，它有3个水平，水平效应为β_j，j=1,2,3。设在显著性水平$\alpha = 0.05$下检验

H_1：$\alpha_1=\alpha_2=\alpha_3=\alpha_4=0$

H_2: $\beta_1=\beta_2=\beta_3=0$

```
>>x=[58.2 56.2 65.3;49.1 54.1 51.6;60.1 70.9 39.2;75.8 58.2 48.7];
  [p,t,st]=anova2(x)
```

得到如下的结果：

```
p = 0.4491    0.7387
t =  'Source'      'SS'          'df'      'MS'         'F'      'Prob>F'
    'Columns'   [    223.8467]    [ 2]    [111.9233]   [0.9174]   [0.4491]
    'Rows'      [    157.5900]    [ 3]    [ 52.5300]   [0.4306]   [0.7387]
    'Error'     [    731.9800]    [ 6]    [121.9967]         []         []
    'Total'     [1.1134e+003]     [11]            []         []         []
```

因p=0.4491 0.7387，表明各种燃料和各种推进器之间的差异对于火箭射程无显著影响。

例 2.35　为提高某种化学产品的转化率（%），考虑三个有关因素：反应温度 A（℃），反应时间 B（min）和使用催化剂的含量 C（%）。各因素选取三个水平。根据正交试验结果，得到表 5.21 所示的结果。请对此进行方差分析。

表 5.21　转化率正交试验结果

因素 / 序号	反应温度A	反应时间B	催化剂含量C	转 化 率
1	80（1）	90（1）	6（2）	31
2	85（2）	90（1）	5（1）	54
3	90（3）	90（1）	7（3）	38
4	80（1）	120（2）	5（1）	53
5	85（2）	120（2）	7（3）	49
6	90（3）	120（2）	6（2）	42
7	80（1）	150（3）	7（3）	57
8	85（2）	150（3）	6（2）	62
9	90（3）	150（3）	5（1）	64

解：

```
>>y=[31 54 38 53 49 42 57 62 64];g1=[1 2 3 1 2 3 1 2 3];g2=[1 1 1 2 2 2 3 3 3];
  g3=[2 1 3 1 3 2 3 2 1];
  [p,t,st]=anovan(y,{g1,g2,g3})
```

得到如下结果：

```
p = 0.1364    0.0283    0.0714
t = 'Source'  'Sum Sq.'  'd.f.'  'Singular?'  'Mean Sq.'   'F'       'Prob>F'
    'X1'      [114.0000]   [2]         [0]    [ 57.0000]  [ 6.3333]  [0.1364]
    'X2'      [618.0000]   [2]         [0]    [309.0000]  [34.3333]  [0.0283]
    'X3'      [234.0000]   [2]         [0]    [117.0000]  [13.0000]  [0.0714]
```

```
    'Error'   [ 18.0000]   [2]      [0]      [   9.0000]      []       []
    'Total'   [      984]  [8]      [0]               []      []       []
```

求得概率p= 0.1364 0.0283 0.0714，可见因素B、C的各水平对指标值的影响有显著差异（显著性水平取0.1），而因素A的各水平对指标值的影响无显著差异。

例 2.36　在对某湖泊水质进行环境监测时，设 15 个监测点，每个监测点监测指标为 5 项（见表 5.22），用主成分分析法确定最佳的监测布设点。

表 5.22　水质监测数据表（单位：mg/l）

点　位	DO	COD	BOD	T-N	T-P
1	4.3	4.74	4.23	3.66	0.105
2	5.9	4.61	2.59	2.92	0.081
3	7.0	3.94	2.92	1.71	0.072
4	6.9	3.92	3.11	1.32	0.075
5	7.4	4.02	3.10	1.26	0.076
6	6.9	3.75	3.15	1.05	0.096
7	6.7	4.44	3.14	1.02	0.072
8	6.8	4.35	4.08	1.27	0.110
9	6.2	4.24	2.33	0.71	0.068
10	7.4	3.99	2.84	0.74	0.063
11	8.1	4.43	3.44	0.86	0.070
12	7.7	4.31	3.50	0.93	0.074
13	5.7	4.88	5.02	1.84	0.134
14	6.8	4.73	4.34	1.39	0.109
15	5.5	5.93	5.06	2.81	0.240

解：

```
>> x=[4.3000  4.7400   4.2300   3.6600   0.1050;5.9000 4.6100   2.5900   2.9200   0.0810;
     7.0000  3.9400   2.9200   1.7100   0.0720;6.9000 3.9200   3.1100   1.3200   0.0750;
     7.4000  4.0200   3.1000   1.2600   0.0760;6.9000 3.7500   3.1500   1.0500   0.0960;
     6.7000  4.4400   3.1400   1.0200   0.0720;6.8000 4.3500   4.0800   1.2700   0.1100;
     6.2000  4.2400   2.3300   0.7100   0.0680;7.4000 3.9900   2.8400   0.7400   0.0630;
     8.1000  4.4300   3.4400   0.8600   0.0700;7.7000 4.3100   3.5000   0.9300   0.0740;
     5.7000  4.8800   5.0200   1.8400   0.1340;6.8000 4.7300   4.3400   1.3900   0.1090;
     5.5000 5.9300   5.0600   2.8100   0.2400];
>>stdr=std(x);sr=x./stdr(ones(15,1),:);
>> [pcs newdata,variances,t2]=princomp(sr);
>> variances'                                  %特征值
ans =3.5195      0.9347    0.2503    0.1686    0.1268
>> (100*variances/sum(variances))'            %特征值贡献率
ans =70.3898     18.6946    5.0061    3.3725    2.5370
```

```
>> pcs(:,1:2)'                                          %前两个主成分
ans =0.4180     -0.4836   -0.4336   -0.4230   -0.4736
    0.5645       0.2255    0.4508   -0.5528    0.3489
```

pcs的值分别代表5项指标在主成分中的权系数，即作用大小。从污染角度出发，根据各指标在主成分中的作用大小，分别给第I、第II主成分赋予物理意义。从pcs的值可看出，在第I主成分中，第1项（DO）对水质的影响是正的，而第2～5项（分别为COD、BOD、T－N、T－P）对水质的影响是负的，主要反映了有机污染物和水质自净作用的对比程度，该值越大，说明水质越好，自净能力强；而第II主成分主要反映环境单元在第I主成分值大体固定的条件下水体中的氨的形成富营养化程度的量度，随着T－N项权值的增加，说明富营养化引起水质的下降。

```
>> newdata(:,1:2)'                                      %主成分的得分
ans =Columns 1 through 11
   -2.7466 -0.4833 1.0960 1.1267 1.2759 1.1648 0.7322 -0.1523 1.3053 1.8233 1.2881
   -2.0751 -1.8049 -0.5834 -0.2815 0.0918 0.0012 -0.0005 0.6667 -0.6494 0.1618
 Columns 12 through 15
     1.1164   -2.1164   -0.6750   -4.7553
     0.7932    0.5853    0.8832    1.1547
>> plot(newdata(:,1),newdata(:,2),'o')       %主成分得分如图5.16所示
```

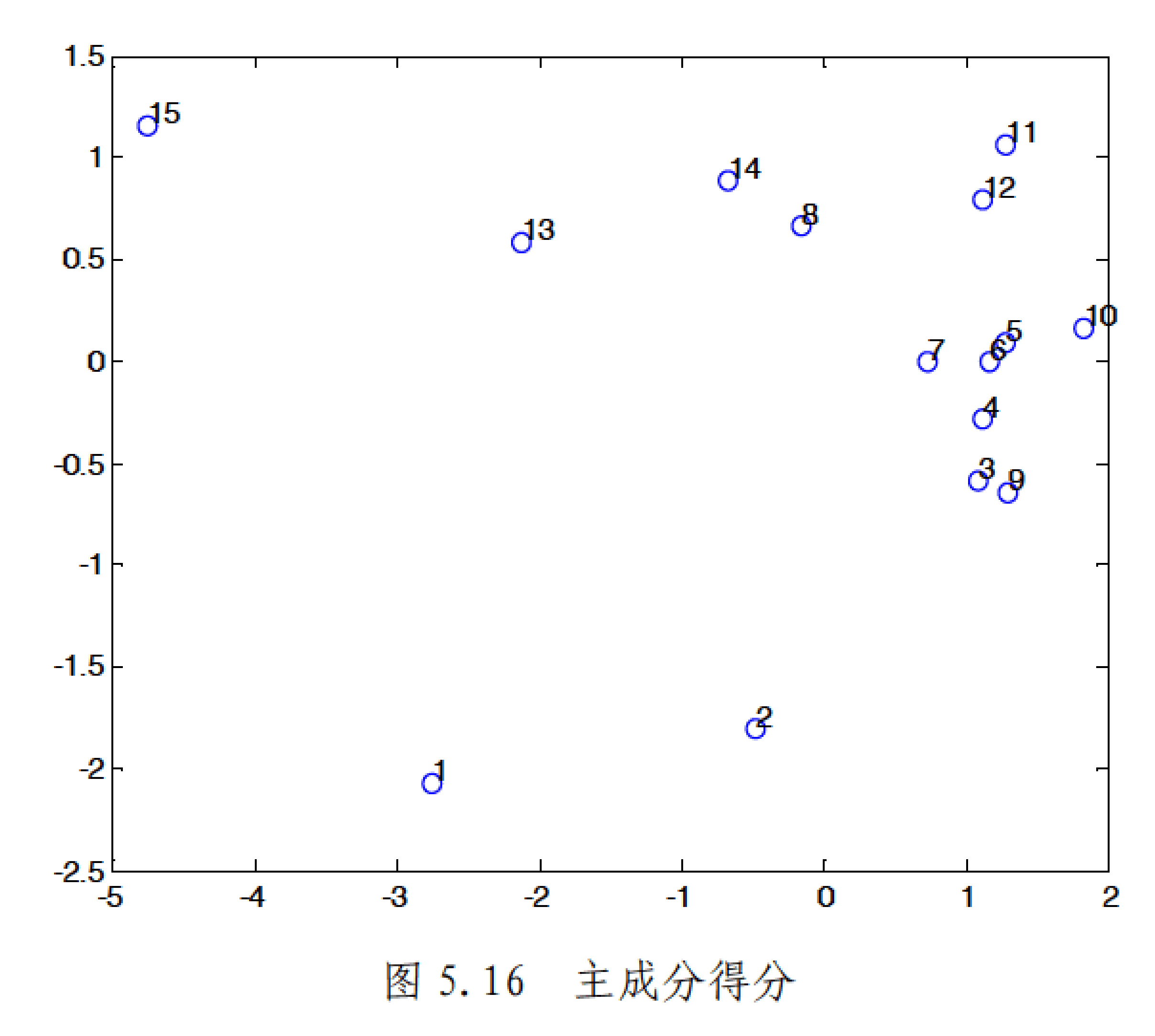


图 5.16　主成分得分

从图5.16中可看出，15个观察点被分成6类：（3,4,5,6,7,9,10,11,12），(8,14)，（1），（2），（13），（15）。而这6类在二维平面图上是按照一定的方向和顺序依次排列的，自右到左，污染程度逐渐增加。不同的污染类别，实质上是客观反映了沿岸工业、人口分布对水环境的影响，两相邻类在污染类型上具有一定的相似性，而在污染程度上具有显著的差异性。

从分类结果看，尚需进一步优选的类别有（3,4,5,6,7,9,10,11,12）、(8,14)两类，可根据类间点位的主成分值相差最大的原则选择，参考得分值，明显最佳点为（10）、(14)。至此，15个观察点经优选后的最佳点位为（1,2,10,13,14,15）。

例 2.37　为了检测某工厂的大气质量情况，在 8 个取样点进行取样并进行分析，得到如表 5.23 所示的分析结果。试对其进行 R 型因子分析。

表 5.23　大气环境质量检测结果　　单位：μg/m

序　号	氯	硫 化 氢	二氧化硫	硫四气体	环氧氯丙烷	环 己 烷
1	0.056	0.084	0.031	0.038	0.0081	0.022
2	0.049	0.055	0.100	0.110	0.022	0.0073
3	0.038	0.130	0.079	0.170	0.058	0.043
4	0.034	0.095	0.058	0.160	0.200	0.029
5	0.084	0.066	0.029	0.320	0.012	0.041
6	0.064	0.072	0.100	0.210	0.028	1.380
7	0.048	0.089	0.062	0.260	0.038	0.036
8	0.069	0.087	0.027	0.050	0.089	0.021

解：

```
>> load mydata;
>> [d,y]=R_factor(x);            %d为因子，y为各因子的得分
>> d= 0.9740     -0.2265
     -0.9828     -0.1846
     -0.2289      0.9735
      0.7305      0.6829
     -0.9775     -0.2110
      0.3415      0.9399
```

从分析结果不难看出：第一主因子主要由氯、硫化氢、环氧氯丙烷和环已烷等构成，而第二主因子由二氧化硫、碳四气体和环已烷等构成，两个主因子体现的污染源不一样。另外从图5.17 中也可以看出各个样本的主要污染物种类。

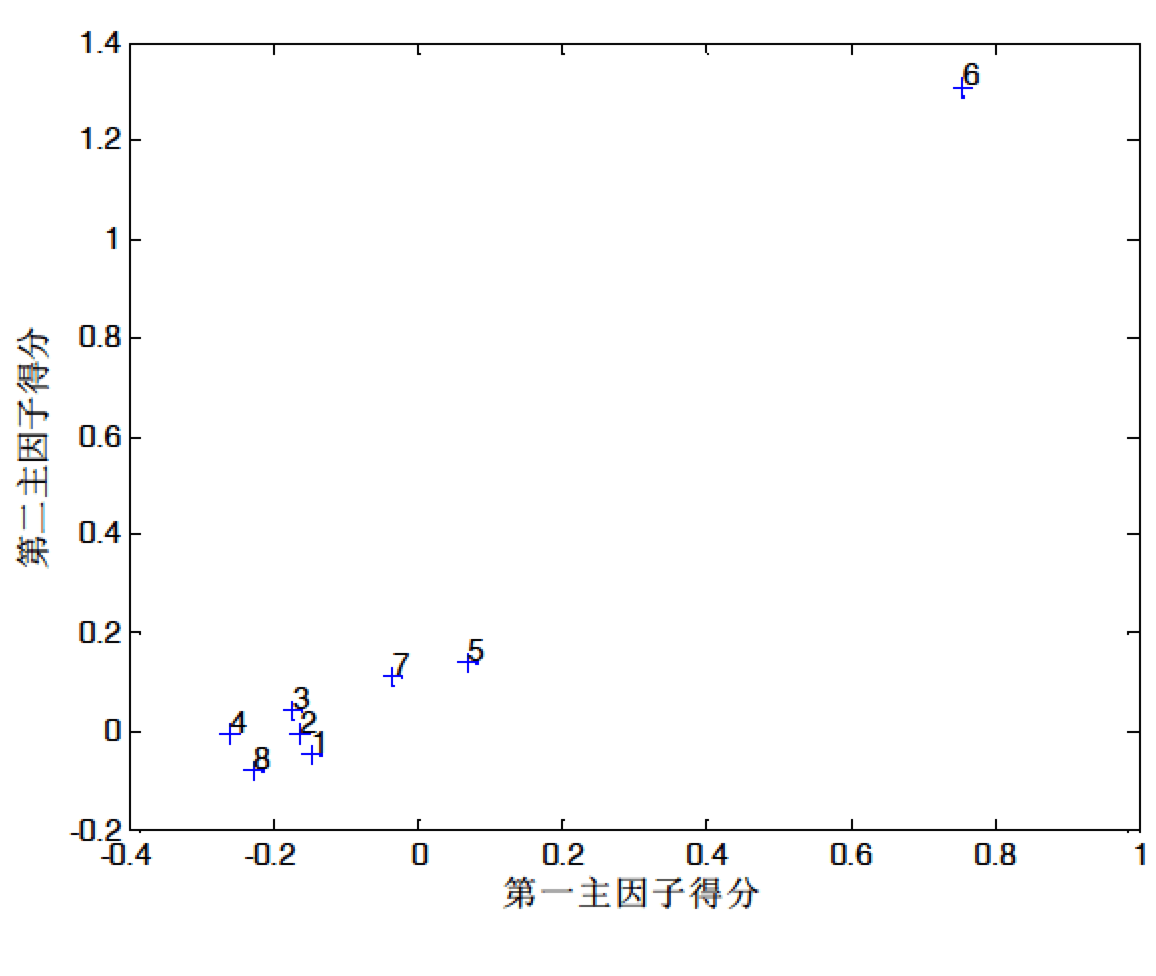


图 5.17　各因子得分

例 2.38　对例 2.36 的数据进行 Q 型因子分析。

解：

Q型因子分析除了输入矩阵不同于R型因子分析，其他运算过程完全一样。

```
>> y=Q_factor(x);    %Q型因子分析函数，限于篇幅不再列出
>> y= 0.9637          -0.2671
      0.9969          -0.0790
      0.9788          -0.2047
      0.9954          -0.0956
      0.9835          -0.1812
      0.1504          -0.9886
      0.9291          -0.1560
      0.8557          -0.1317
```

例 2.39　在 MATLAB 中因子分析的极大似然估计函数为 Factoran，其调用格式为

```
[lambda, psi, t, stats] =factoran(X, M)
```

其中：X 是观察向量；M 是公共因子的数目；psi 返回的特殊因子负荷矩阵的估计值；*t* 返回因子负荷旋转矩阵；stats 是一个数据结构，包含了与假设统计检验有关的信息。详细调用格式见该函数的 help。

对例 2.35 的数据，用 factoran 函数分析之，以确定它最佳的监测布设点。

解：

```
>> [a,b,c,d,f]=factoran(x,2,'rotate','promax');    %因子数设为2，利用最大方差旋转载荷阵
```

利用结果可分别作图 5.18 和图 5.19，从图中可看出，2、3、5 指标则与因子 1 有关，1、4 指标与因子 2 有关。15 个观察点被分成 6 类：(3,4,5,6,7,9,10,11,12)，(8,14)，(1)，(2)，(13)，(15)。

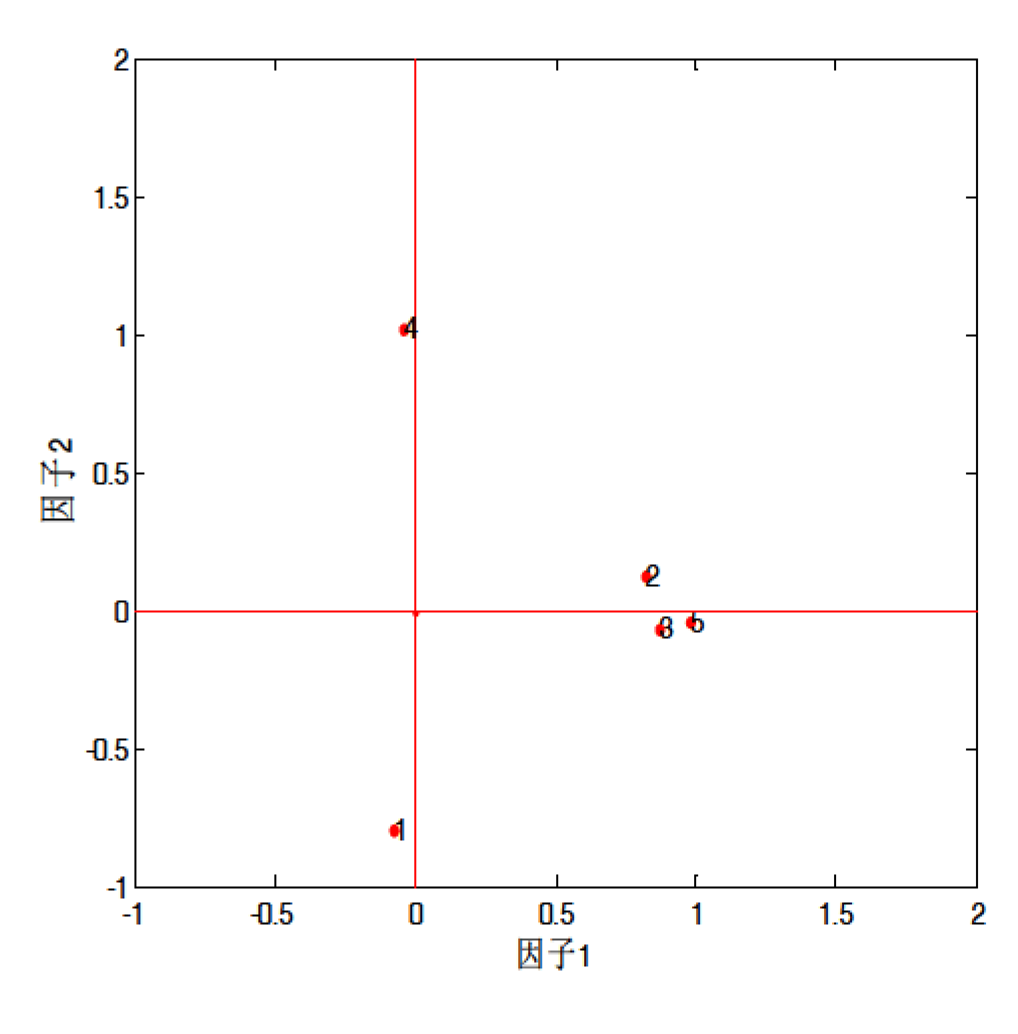


图 5.18　因子图

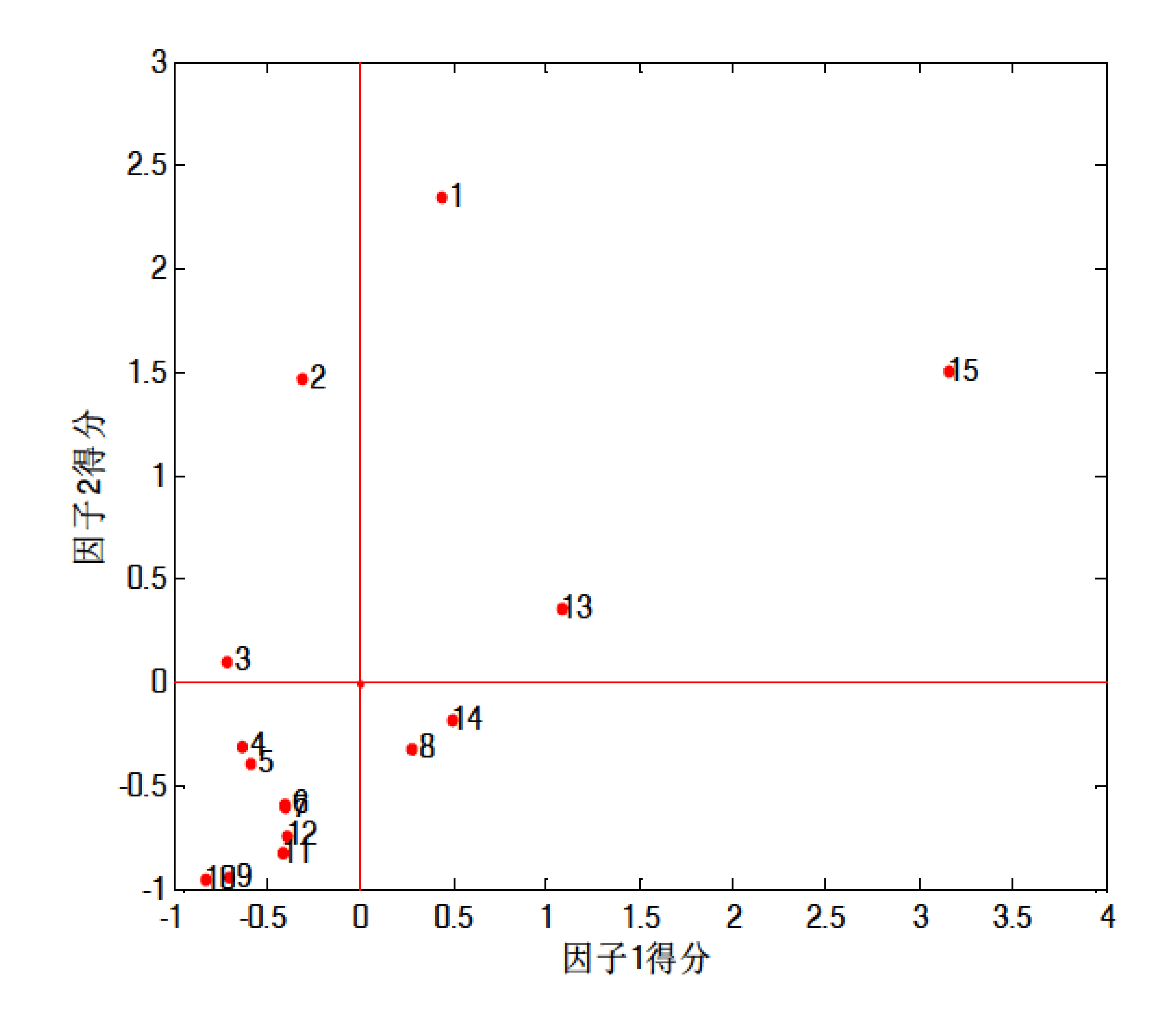


图 5.19　因子得分

从分类结果看，尚需进一步优选的类别有（3,4,5,6,7,9,10,11,12）和(8,14)两类，可根据类间点位的载荷分值相差最大的原则选择，参考得分值，明显最佳点为（10）、(14)。至此，15 个观察点经优选后的最佳点位为（1,2,10,13,14,15）。

例 2.40　在制定服装标准的过程中，对 128 名成人的身材进行了测量，每人测了身高、坐高、胸围、手臂长、肋围和腰围 6 项指标，其数据样本的相关矩阵如表 5.24 所示。试对表中数据进行因子分析。

表 5.24　128 名成人身材 6 项指数数据的相关系数矩阵

序　号	身　高	坐　高	胸　围	手 臂 长	肋　围	腰　围
1	1	0.79	0.36	0.76	0.25	0.51
2	0.79	1	0.31	0.55	0.17	0.35
3	0.36	0.31	1	0.35	0.64	0.58
4	0.76	0.55	0.35	1	0.16	0.38
5	0.25	0.17	0.64	0.16	1	0.63
6	0.51	0.35	0.58	0.38	0.63	1

解：

对相关系数矩阵进行因子分析，设公共因子为 2，特殊方差的下限为 0

```
>>x=[1 0.79 0.36 0.76 0.25 0.51;0.79 1 0.31 0.55 0.17 0.35;0.36 0.31 1 0.35 0.64 0.58;0.76 0.55 0.35 1 0.16 0.38;0.25 0.17 0.64 0.16 1 0.63;0.51 0.35 0.58 0.38 0.63 1];
>> [lamda,psi,T]=factoran(x,2,'xtype','covariance','delta',0,'rotate','none');
>>head={'变量','因子 f1','因子 f2'};
>> varname={'身高','坐高','胸围','手臂长','肋围','腰围','贡献率(%)','累积贡献率(%)'}';
```

```
>> contribut=100*sum(lamda.^2)/6;cumcont=cumsum(contribut);
>> s1=num2cell([lamda;contribut;cumcont]);
>> s=[head;varname,s1]
s =  '变量'              '因子 f1'        '因子 f2'
     '身高'              [ 1.0000]      [-2.1074e-006]
     '坐高'              [ 0.7900]      [     -0.0292]
     '胸围'              [ 0.3600]      [      0.6573]
     '手臂长'            [ 0.7600]      [ 3.4051e-004]
     '肋围'              [ 0.2500]      [      0.8355]
     '腰围'              [ 0.5100]      [      0.6026]
     '贡献率(%)'         [44.2317]      [     24.8999]
     '累积贡献率(%)'     [44.2317]      [     69.1316]
```

由变量在因子上的载荷可以看出，因子 1 反映的是身高、坐高和手臂长，称为身高因子；因子 2 反映的是胸围、肋围和腰围，说明的是胖瘦，称为胖瘦因子。

从特殊方差 psi 来看，第一个特殊方差达到了参数 delta 的取值 0，表明出现了海伍德现象，而其他方差比较大，再考虑到前两个因子的贡献率只有 69%左右，说明拟合不足，可以考虑增加因子数目。

下面利用最大方差旋转法对因子进行旋转，观察因子载荷的变化：

```
>> [lamda,psi,T]=factoran(x,2,'xtype','covariance','delta',0)
>> contribut=100*sum(lamda.^2)/6;cumcont=cumsum(contribut)
```

可以看到因子旋转后，旋转矩阵 T 发生了变化， 并且因子载荷每列上的各元素差异更明显，更容易对因子做出解释了，但是因子的累积贡献率没有变化。

下面增加因子数，观察因子载荷的变化情况：

```
>> [lamda,psi,T]=factoran(x,3,'xtype','covariance','delta',0)
>> contribut=100*sum(lamda.^2)/6;cumcont=cumsum(contribut)
```

此时仍没有消除海伍德现象，从特殊方差来看，第 4 个变量（手臂长）的特殊方差为 0.0132，说明它也得到了很好的拟合，但是其他变量的特殊方差还是较大，拟合仍不足。由于受$(d-m)^2 \geq d+m$（维数 d=60）的限制，因子数不能继续增大。事实上即使再增加因子数，虽然能消除海伍德现象，但因子也失去了作为公共因子的意义，并且解释时也可能会变得困难，这也是毫无意义的。

例 2.41　典型相关分析是分析变量间关系的一种常用方法，它分别从两组数据中提取相关性最大的两个成分，通过测定这两个成分之间的相互关系，来推测两个数据间的相互关系。典型相关分析有着重要的应用背景，如在宏观经济分析中，研究国民经济的投入要素与产出要素这两组变量间的联系情况；在市场分析中，研究销售情况与产品性能间的关系等。

试对表 5.25 的某矿床数据集进行典型相关分析，以揭示矿床的成因。

表 5.25　某矿床数据集

岩　体				矿　体	
Al_2O_3	Fe_2O_3	Na_2O	K_2O	Cu	S
17.71	0.65	3.12	3.36	0.99	3.14
16.31	0.25	3.27	2.89	1.19	10.30
18.07	0.58	3.66	1.85	0.71	11.78
16.67	0.49	3.69	1.80	1.35	5.01
17.10	0.20	3.39	2.09	1.36	1.78
17.57	0.66	3.98	1.92	0.95	3.01
17.70	0.62	4.40	1.45	0.87	2.99
19.24	0.82	4.00	1.94	0.75	4.86
17.97	1.41	4.07	2.10	0.61	2.87
17.89	0.83	4.15	2.05	1.13	14.31

解：

```
 >>x1=[17.71 0.65 3.12 3.36 0.99 3.14;16.31 0.25 3.27 2.89 1.19 10.30;
       18.07 0.58 3.66 1.85 0.71 11.78;16.67 0.49 3.69 1.80 1.35 5.01;
       17.10 0.20 3.39 2.09 1.36 1.78;17.57 0.66 3.98 1.92 0.95 3.01;
       17.70 0.62 4.40 1.45 0.87 2.99;19.24 0.82 4.00 1.94 0.75 4.86;
       17.97 1.41 4.07 2.10 0.61 2.87;17.89 0.83 4.15 2.05 1.13 14.31];
>>x=x1(:,1:4);y=x1(:,5:6);
>> [A,B,r,U,V,stats] = canoncorr(x,y);    %典型相关分析函数
>> A= -0.7073   -0.4147
      -1.8243    3.6691
       0.3575   -5.2949
       0.1564   -3.1797
>> B = 3.7912    0.3310
       0.0104   -0.2254
>> r = 0.8240    0.1909                     %典型相关系数
```

矿体的元素 Cu、S 与地层的化学成分 Al_2O_3、Fe_2O_3、Na_2O、K_2O 间关系密切，特别是矿床的主要元素 Cu 与地层中的 Fe_2O_3 关系密切。从地质统计学方面来讲，铜矿床为与地层有关的热水沉积矿床。

例 2.42　对我国国家统计局网站提供的 2007 年我国大陆地区 31 个省、自治区、直辖市的农村居民家庭平均每人全年消费性支出的数据表进行主成分分析。

解：

数据表中有食品、衣着、居住、家庭设备及服务、医疗保健、交通和通信、教育文化娱乐服务，杂项商品服务共 8 个变量。通过对数据的主成分分析，可以得知各地消费水平及其特点。

在主成分分析中，最为关键的是确定应提取多少个主成分。提取多少个主成分的标准是：①特征值标准：只有特征值大于 1 的主成分应予保留。②解释变异比例标准：通过指定期望主成分能解释总变异的量，然后一个接一个地选择要素，直到达到理想的解释变异比例。例如假设希望主成分解释 85%的变异量。③最小共性指标：共性代表了各变量在主成分分析中的总体重要性。较高的共性值（即得分矩阵中的系数）表示主成分成功地提取了初始变量中的大部分波动；较低的共性值说明数据集中仍有一些未被主成分所解释的波动。④碎石图：碎石图是特征值与主成分数量的关系图，它可用于寻找一个上限（最大值）来决定多少主成分应予以保留。大部分碎石图在形状上大致相同，左侧开始处高，迅速下降，然后从某一点开始变平坦。一般曲线开始趋于平坦的分界点便是最多主成分数量的取值。

根据以上要点，就可以对给出的数据进行主成分分析。

```
>> varname={'食品'  '衣着'  '居住'  '家庭设备及服务'  '医疗保健'   '交通和通信'  '教育文化娱乐服务'   '杂项商品服务'};
>>samplename={'北京','天津','河北','山西','内蒙古','辽宁','吉林','黑龙江','上海','江苏','浙江','安徽','福建','江西','山东','河南','湖北','湖南','广东','广西','海南','重庆','四川','贵州','云南','西藏','陕西','甘肃','青海','宁夏','新疆'};
>>load mydata;
>> [m1,m2,m3,m4]=princomp_analy(x,0.95,varname,samplename);   %主成分分析函数
可以得到以下的分析结果：
s1= '特征值'       '差值'       '贡献率'     '累积贡献率'
     [6.8649]    [6.2904]    [0.8581]    [    0.8581]
     [0.5746]    [0.4060]    [0.0718]    [    0.9299]
     [0.1686]    [0.0236]    [0.0211]    [    0.9510]
     [0.1449]    [0.0464]    [0.0181]    [    0.9691]
     [0.0986]    [0.0149]    [0.0123]    [    0.9814]
     [0.0837]    [0.0403]    [0.0105]    [    0.9919]
     [0.0435]    [0.0222]    [0.0054]    [    0.9973]
     [0.0213]          []    [0.0027]    [         1]
```

由 s1 矩阵及图 5.20 所示的碎石图可以确定主成分数为 2。

```
>> m2 为各主成分与变量间的关系：
   m2= '标准化变量'          '主成分 1'    '主成分 2'
        '食品'               [ 0.3432]   [ 0.5030]
        '衣着'               [ 0.3384]   [-0.4869]
        '居住'               [ 0.3552]   [ 0.1966]
        '家庭设备及服务'     [ 0.3692]   [ 0.1089]
        '医疗保健'           [ 0.3751]   [-0.0526]
        '交通和通信'         [ 0.3587]   [-0.2212]
```

```
      '教育文化娱乐服务'     [ 0.3426]     [-0.4791]
      '杂项商品服务'         [ 0.3442]     [ 0.4221]
```

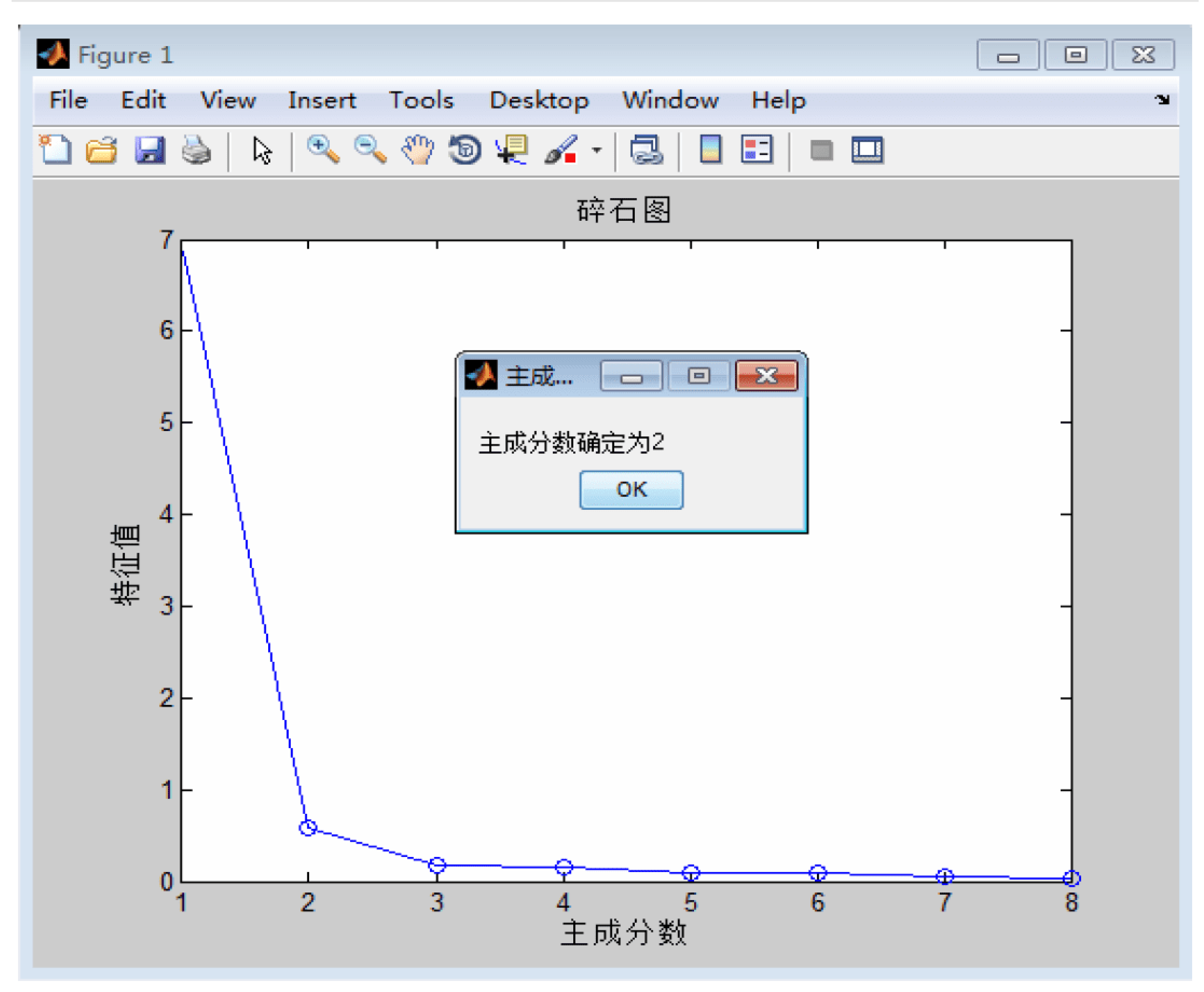


图 5.20 碎石图

可以看出，主成分 1 中的每个变量都有相近的载荷，说明每个标准化变量对主成分 1 的重要性相差不大，它反映的应该是综合性消费支出水平。而主成分 2 在食品与杂项商品服务上有中等程度的负载荷，在衣着和医疗保健上有中等程度的正载荷，说明它反映的是两个方面的对比，一方面是衣着和医疗的消费总支出，另一方面是食品和其他商品及服务的消费总支出，所以可以认为主成分 2 反映的是消费倾向成分。这个结论可以从 m1 中的两个矩阵数据得到证实。在计算过程中，因为主成分 1 反映的是总支出，所以在计算各地区总消费支出水平 m1{1}的情况时是以主成分 1 为基准（即对主成分 1 排序）；在计算各地区消费倾向（m1{1}）时是以主成分 2 为基准（即对主成分 2 排序）。

后几个主成分的贡献率较小，可以不做解释，但却说明了标准化变量之间可能存在一个或多个共线性关系。

另外，根据图 5.21 两个主成分得分散点图可以把 31 个地区分为 3 类，其中北京、浙江和上海为第一类，江苏、福建和广东为第二类，其余为第三类。

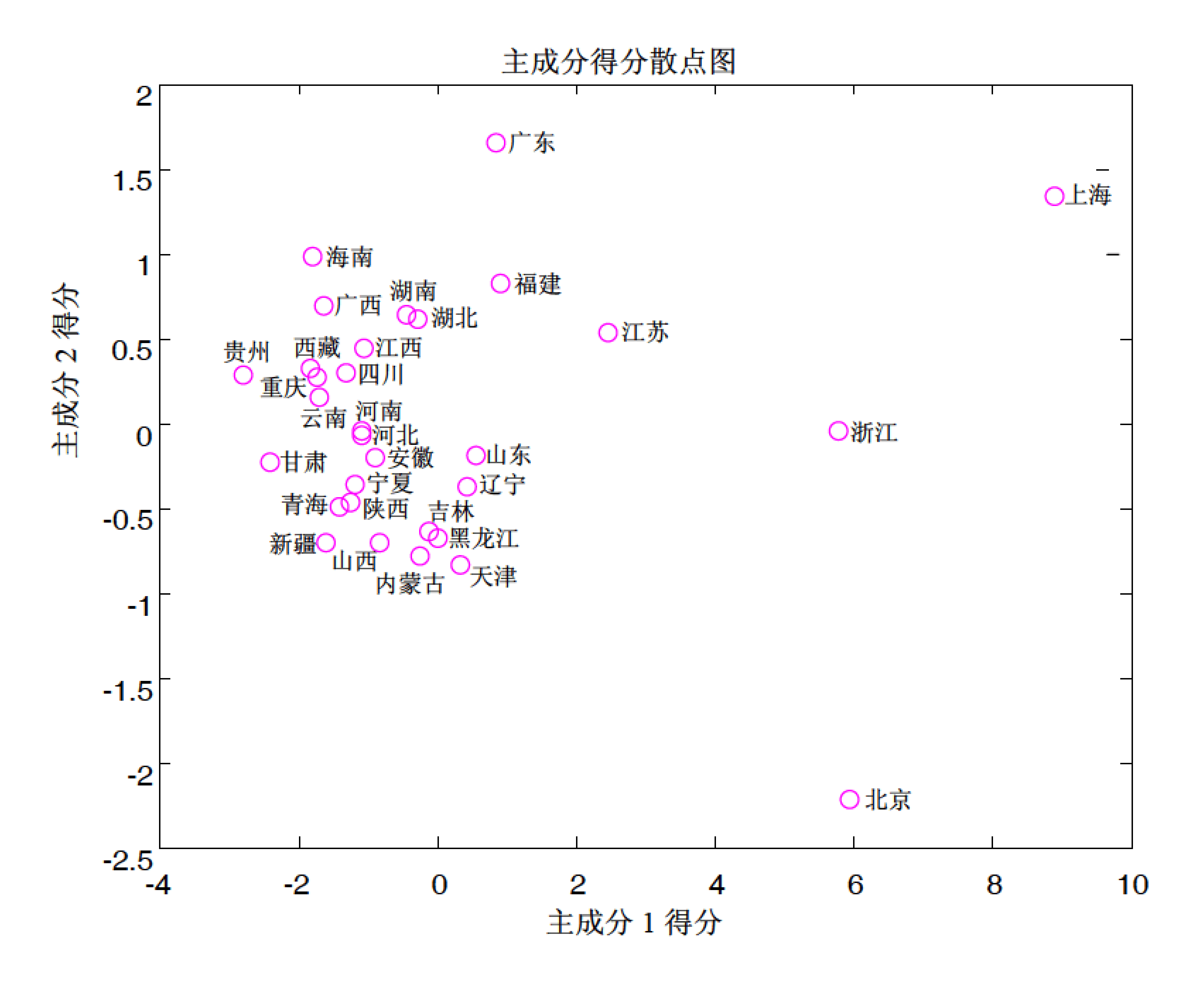


图 5.21　前两个主成分得分的散点图

m3 说明各变量的权重。从权重矩阵也可以确定较为适宜的主成分数量。

m4 的数据可以说明数据集中聚簇情况，在本例中上海是距离数据集中最远的地区。

第 6 章

贝叶斯网络方法

6.1 贝叶斯定理、先验和后验

贝叶斯理论是一种研究不确定性的推理方法。

不确定性常用贝叶斯概率表示，它是一种主观概率。通常的经典概率代表事件的物理特性，是不随人的意识变化的客观存在，而贝叶斯概率则是人的认识，是个人主观的估计，随个人的主观认识的变化而变化。如在投掷硬币的实验中，贝叶斯概率是指个人相信硬币会正面向上的程度。

主观概率不像经典概率那样强度多次的重复，因此在许多不可能出现重复事件的场合能得到很好的应用，如投资者对股票是否能取得高收益的预测都不可能进行重复的实验。

因此利用主观概率，按照个人对事件的相信程度而对事件做出推断是一种很合理而易于解释的方法。

在贝叶斯理论之上可以建立贝叶斯网络。贝叶斯网络是用来表示变量之间连接关系概率的图形模式，它提供了一种自然的表示因果关系的方法，刻画了信任度与证据的一致性以及信任度随证据而变化的增量学习特性，以概率的权重来描述数据间的相关性。

使用 $p(X=x|\mathrm{A})$或者 $p(x|\mathrm{A})$表示给定知识 A 的情形下对事件 $X=x$ 的相信程度，即贝叶斯概率，它同时也是 X 的分布（或分布密度）。

如果 θ 是一个参数，$p(\theta|\mathrm{A})$表示在给定知识 A 的前提下 θ 的分布，$D=\{X_1=x_1,\cdots,X_N=x_N\}$ 表示观测数据集合，则 $p(\theta|D，\mathrm{A})$表示给定知识 A 和数据 D 时参数 θ 的分布，其中 $p(\theta|\mathrm{A})$表示参数 θ 的先验密度，有知识 A 表示该先验不是无知识先验，它是在掌握知识 A 后给出的先验密度，$p(\theta|D，\mathrm{A})$表示参数 θ 的后验密度，它是在已知知识 A 和数据 D 之后对参数的分布密度的估计。在实际表示中，可以省略知识 A。

由贝叶斯法则有

$$p(\theta\,|\,D)p(D)=p(\theta,D)=p(\theta)p(D\,|\,\theta)$$

经过简单变化，可以得到由先验和数据计算后验的贝叶斯定理

$$P(\theta\,|\,D)=\frac{P(D\,|\,\theta)P(\theta)}{\int P(D\,|\,\theta)P(\theta)}$$

式中的 $p(\theta|D)$常常被称为似然函数，用 $l(\theta|D)$表示，此时贝叶斯定理常可表示为

$$P(\theta\,|\,D)\propto l(D\,|\,\theta)P(\theta)$$

一般来说，先验分布反映人们在数据获得之前对参数（或其他概率知识）的认识；后验则是反映在获得数据之后对参数的认识。

两者的差异是由于数据出现后对参数的一种调整。所以从这个角度看，先验和后验是相对的，当需要利用新数据更新参数的分布密度时，已知的参数分布密度就是先验分布密度，更新后的参数分布就是后验分布。

这一更新过程可以重复进行，只要有新的数据信息，就可以根据贝叶斯定理对先验分布密度进行更新，得到后验分布密度。

贝叶斯定理给出了一种根据新数据不断更新后验分布的序贯方法。如果获得了新的数据集合 D^*，则在获得数据 D 和 D^*后参数的后验分布为

$$P(\theta\,|\,D^*,D)=\frac{P(D^*\,|\,\theta)P(\theta\,|\,D)}{P(D^*)}$$

6.2　贝叶斯网络

设U是一个随机变量，$U=\{X_1,X_2,\cdots,X_n\}$，其中X_i从一有限集Val(X_i)中取值。U的一个贝叶斯网络定义了U上的一个联合概率分布。

以B=〈G,Θ〉表示一贝叶斯，其中G是一个有向无环图，其顶点对应于有限集U中的随机变量$X_1,X_2,\cdots,X_n$，其弧代表一个函数依赖关系，如果有一条弧从X_i到X_j，则X_i是X_j的双亲或直接前驱（或父节点），X_j是X_i的后继（或子节点），变量X_k所有双亲变量用集合Pa(X_k)表示，并用pa(X_k)表示Pa(X_k)的一个取值。一旦给定其双亲，图中的每个变量独立于图中该连接点的非后续。

这里的独立是指条件独立，其定义是：给定Z，X_i，X_j是条件独立的，如果$\forall x_i\in X_i,\forall X_j\in X_j$，$\forall z\in Val(Z)$，当P($X_j$,$Z$)>0时，有$P(x_i\mid z,x_j)=P(x_i\mid z)$成立。其中Θ表示用于量化网络的一组参数，对于每一个X_i的取值x_i，以及Pa(X_k)的pa(X_k)取值，存在一个参数，$\theta_{i|pa(X_i)}=P(x_i\mid pa(X_i))$，指明了在给定pa($X_k$)下$x_i$发生的条件概率。图6.1即为一个贝叶斯网络。

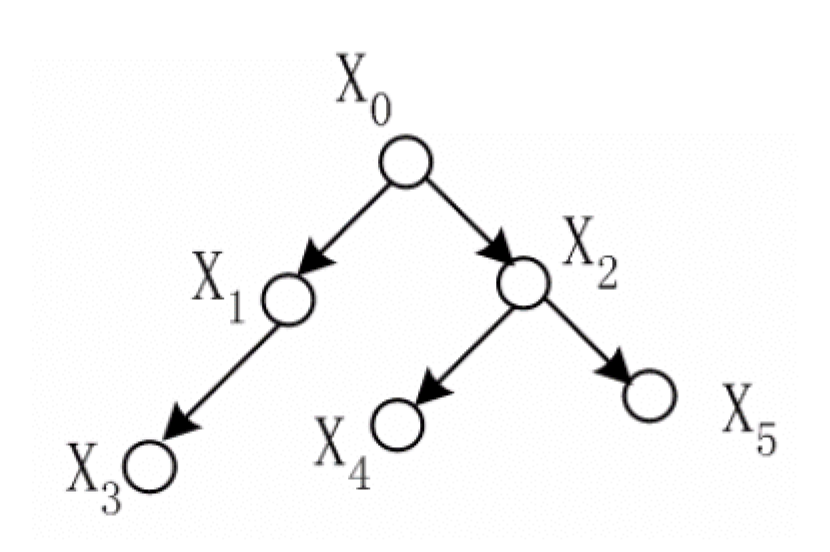


图6.1　一个贝叶斯网络

实际上贝叶斯网络给定了变量集合X上的联合条件概率分布

$$p(X_1,X_2,\cdots,X_n)=\prod_{i=1}^{n}p(X_i\mid pa(x_i))$$

贝叶斯网络的建立主要有两个相继环节：一个是结构学习；另一个是参数学习。

① 结构学习是利用一定的方法建立贝叶斯网络结构的过程，该过程决定了各个变量间的关系，结构学习环节是贝叶斯网络分类算法的最重要的步骤，是参数学习环节与分类环节的基础。

② 参数学习是量化网络的过程，它在网络结构已知的情况下计算各节点X_i的条件概率。

通常用以下三种不同的方式来构造贝叶斯网络。

（1）由领域专家确定贝叶斯网络的变量（有时也称为影响因子），然后通过专家的知识来确定贝叶斯网络的结构，并指定它的分布参数。这种方式构造的贝叶斯网络完全在专家的指导下进行，由于人类获得知识的有限性，导致构建的网络与实践中积累下的数据有较大偏差。

（2）由领域专家确定贝叶斯网络的特点，通过大量的训练数据，来学习贝叶斯网络的结构与参数。这种方法完全是一种数据驱动的方法，具有很强的适应性，而且随着人工智能、数据挖掘和机器学习的不断发展，使得这种方法更加普及。

（3）由领域专家确定贝叶斯网络的特点，通过专家的知识来指定网络的结构，而通过机器学习的方法从数据中学习网络的参数。这种方法实际是前两种方法的折中，当领域中变量之间的关系较为明显的情况下，这种方法能大大提高学习的效率。

6.3 贝叶斯网络学习

6.3.1 贝叶斯网络的结构学习

贝叶斯网络结构学习算法主要分析节点依赖关系与节点连接关系。常用的方法是基于评分–搜索的贝叶斯网络结构学习和基于信息化的依赖分析方法。

（1）基于评分搜索的贝叶斯网络结构学习的算法将学习问题看作是数据集寻找最合适的结构，这类算法从没有边的图形开始，利用搜索方法将边加入到图形中。然后，利用测试方法检验是否新的结构优于旧的结构。如果是，保存新加上的边并继续加入其他边。这个过程一直持续到最优的结构。不同的测试标准可以应用在算法中以评价结构的优劣。大多数算法应用的是启发式搜索的方法。为了减少搜索空间，许多算法事先指定结构的次序。

由于该算法需要随着变量增加其运算复杂性，所以当变量较大时，贝叶斯网络结构空间是相当大的，这会使搜索用时较长且结果较差，这导致了准确有效地找到贝叶斯网络分类器的最优网络结构是非常困难的。

（2）基于信息论学习贝叶斯网络的算法主要根据变量之间的依赖性建立贝叶斯网络结构。依赖关系通过变量的相互信息程度定义，如果对应变量的网络节点为X_i和X_j，则X_i和X_j的相互信息可以表示为

$$I(X_i,X_j)=\sum_{X_i,X_j}P(X_i,X_j)\lg\frac{P(X_i,X_j)}{P(X_i)P(X_j)}$$

条件相互信息为

$$I(X_i,X_j\mid C)=\sum_{X_i,X_j,C}P(X_i,X_j,C)\lg\frac{P(X_i,X_j\mid C)}{P(X_i\mid C)P(X_j\mid C)}$$

C是一个节点集合，如果$I(X_i,X_j)\leqslant\varepsilon$（$\varepsilon$ 是一个定值），则节点X_i和X_j依赖较少。

6.3.2 贝叶斯网络的参数学习

贝叶斯网络的参数学习实质上是在已知网络结构的条件下，通过样本学习获取每个节点的概率分布表。初始的贝叶斯网络的概率分布表一般由专家根据先验知识指定，称为网络的先验参数。这样的先验参数可能导致与观察数据产生较大的偏差。要使偏差减少，必须从样本数据中学习以获取更准确的参数及其相应的概率分布。针对完整与不完整数据，贝叶斯网络的参数学习也分为两种不同情况。

1. 基于完整数据的贝叶斯网络参数学习

对完整数据集D进行条件概率学习的目标是找到能以概率形式P（$x|\theta$）概括样本D的参数θ。参数学习一般要首先指定一定的概率分布族，然后采用最大似然估计MLE方法或贝叶斯方法估计这些参数值，下面简单介绍贝叶斯方法。

设定 $X=(X_1,X_2,\cdots,X_n)$ 为对应各节点的随机变量集，B表示贝叶斯网络的结构，θ表示各节

点条件概率分布的随机变量。样本数据 $D=(C_1,C_2,\cdots,C_n)$，每个都是随机变量的实例，目的是通过对样本数据的学习，得到各节点的条件概率分布。

贝叶斯方法学习条件概率由两部分组成，即观察前的先验知识和观测得到的数据。假设参数的先验分布为Dirichlet分布，即

$$P(\theta)=\mathrm{Dir}(\alpha_1,\alpha_2,\cdots,\alpha_n)=\frac{\Gamma(\alpha)}{\prod_i \Gamma(\alpha_i)}\prod_i \theta^{\alpha_i-1}$$

式中：$\alpha=\sum_{i=1}^{N}\alpha_i$ 是分布精度，区别于分布参数，α_i（i=1,…,n）为超参数，这些参数为每个取值出现个数的先验知识。当N=2时为Beta分布，那么样本发生的概率为

$$P(D)=\frac{\Gamma(\alpha)}{\Gamma(\alpha+n)}\prod_i\frac{\Gamma(\alpha_i+n_i)}{\Gamma(\alpha_k)}$$

参数的后验概率也为Dirichlet分布，即

$$P(\theta\mid D)=\frac{P(\theta)P(D\mid\theta)}{P(D)}=\frac{\Gamma(\alpha+n)}{\prod_i\Gamma(\alpha_i+n_i)}\prod_k\theta^{\alpha_i}=\mathrm{Dir}(\alpha_1+n_1,\cdots,\alpha_N+n_N)$$

式中：n_i是训练样本中的x_i第i个值出现的次数，n为总的出现次数。

对于含有多个父节点条件概率计算，θ_{ijk}表示π_j=j时，x_i=k的条件概率，r_i表示x_i的取值个数，q_i表示所有父节点的状态总数，那么在以上假定的基础上，对于每个变量x_i和它的父状态π_j=j服从Dirichlet分布，即

$$P(\theta_{ij1,}\cdots,\theta_{ijn}\mid\zeta)=\zeta\prod_k\theta_{ijk}^{\alpha_{ijk}}$$

在数据集D下的后验分布仍为Dirichlet分布，所以可以用下式来计算条件概率，即

$$\theta_{ijk}=\frac{\alpha_{ijk}+n_{ijk}}{\alpha_{ij}+n_{ij}}\quad(\alpha_{ij}=\sum\alpha_{ijk},n_{ij}=\sum n_{ijk})$$

2. 不完整数据下的贝叶斯网络参数学习

当训练样本集不是完整的情况下，一般要借助近似方法，目前常采用的是Gibbs抽样算法（Gibbs sampling）和EM（Expectation Maximization）算法。

Gibbs抽样算法是一种随机的方法，能近似出变量的初始概率分布，算法定义为：按照候选假设集合H上的后验概率分布，从H中随机选择假设h，使得来预言下一个实例的分类。算法分为三个步骤：首先，随机地对所有未观察变量的状态进行初始化，由此可得出一个完整的数据集；其次，基于这个完整的数据集，对CPT（条件概率表）进行更新；最后，基于更新的CPT参数，用Gibbs抽样算法对所有丢失的数据进行抽样，又得到一个完整的数据集。直到CPT达到稳定时，完成学习过程。

EM算法可用于变量的值从来没有被直接观察到的情形，只要这些变量所遵循的概率分布的一般形式已知即可，可利用EM算法搜索参数的极大后验概率。这个算法包括两个步骤，期望（Expectation Step）和最大化（Maximization Step）。Expectation（E）步骤：用现有参数来估计未观察参数；Maximization（M）步骤：利用估计参数进行参数的ML/MAP估计，将估计值赋给

参数。重复EM步骤，直至收敛。在E步骤，所有节点的期望值可以用推理算法进行计算。其基本思想是：首先给整个网络的CPT选择随机值，并将其作为当前假设g，利用网络结构的CPT做概率推理，得到隐藏变量的概率权重（给定观察数据值时缺失数据集的条件概率），通过采样获得这些变量的估计值，然后利用这些估计值计算出新的最大可能的假设g'，用g'替换。重复以上过程，该过程伴随着隐藏变量的估计值，收敛于本地最大可能的假设，即最大可能的条件概率表。

6.4 主要贝叶斯网络模型

根据变量关系要求的不同，贝叶斯网络一般可分为有约束贝叶斯网络和无约束贝叶斯网络。有约束贝叶斯网络要求变量对应的节点是相互独立或有少量的节点是不独立的，这样的假设可以使网络建立过程的结构简化或参数学习计算量大大减少；而无约束贝叶斯网络允许变量节点是不独立的。

下面介绍几种主要的贝叶斯网络模型。

6.4.1 朴素贝叶斯网络

朴素贝叶斯网络是典型的有约束贝叶斯网络。朴素贝叶斯网络有如图6.2所示的简单结构。这个网络描述了朴素贝叶斯分类器的假设，即给定类变量（网络中的根节点）的状态，每个属性变量（网络中每个叶节点）与其余的属性变量是独立的。

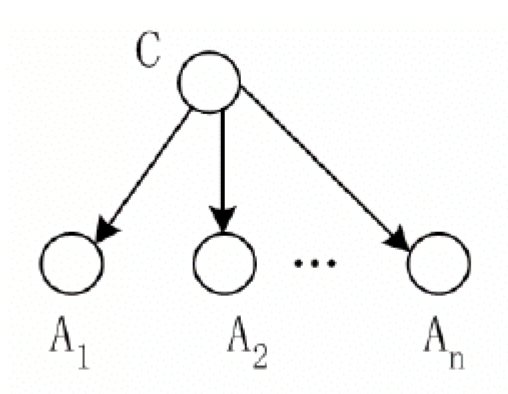


图6.2 朴素贝叶斯网络

朴素贝叶斯网络分类器的工作过程如下：

（1）每个数据样本用一个n维特征向量$X=(X_1,X_2,\cdots,X_n)$表示，分别描述对n个属性$A_1,A_2,\cdots,A_n$样本的n个度量。

（2）假定有m个类$C_1,C_2,\cdots,C_m$。给定一个未知的数据样本X（即没有类标号），分类法将预测X属于具有最高后验概率（条件X下）的类。即朴素贝叶斯分类器将未知的样本分配给类C_i，当且仅当

$$P(C_i|X)>P(C_j|X),1\leqslant j\leqslant m,j\neq i$$

这样，最大化$P(C_i|X)$，使$P(C_i|X)$最大的类C_i称为最大后验假定。根据贝叶斯定理有

$$P(C_i|X)=\frac{P(X|C_i)P(C_i)}{P(X)}$$

（3）由于$P(X)$对于所有类为常数，只需要$P(X|C_i)P(C_i)$最大即可。如果类的先验概率未知，则通常假设这些类是等概率的，即$P(C_1)=P(C_2)=\cdots=P(C_m)$，并据此只对$P(X|C_i)$最大化。否则，最大化$P(X|C_i)P(C_i)$。其中类的先验概率可以用$P(C_i)=s_i/s$计算，其中$s_i$是类$C_i$中的训练样本数，$s$是训练样本总数。

（4）给定具有许多属性的数据集，计算 $P(X \mid C_i)$ 的开销可能非常大。为降低此开销，可以做类条件独立的朴素假设。给定样本的类标号，假设属性值相互条件独立，即在属性之间不存在依赖关系，这样有

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

概率 $P(x_1 \mid C_i), P(x_2 \mid C_i), \cdots, P(x_n \mid C_i)$ 可以由训练样本估值，其中

① 如果A_k是离散属性，则 $P(x_k \mid C_i) = s_{ik} / s_i$，其中$s_{ik}$是在属性$A_k$上具有值$x_k$的类$C_i$的训练样本数，$s_i$是类$C_i$的训练样本数。

② 如果A_k是连续属性，则离散化该属性。

（5）为对未知样本X分类，对每个类C_i，计算 $P(X \mid C_i)P(C_i)$。样本X被指派到类C_i，当且仅当

$$P(C_i \mid X)P(C_i) > P(X \mid C_j)P(C_j), 1 \leqslant j \leqslant m, j \neq i$$

即X被指派到使 $P(X \mid C_i)P(C_i)$ 最大的类C_i。

朴素贝叶斯网络分类器的特点：

- 优点：网络结构非常简单，建立网络时间少，参数学习与分类过程简便。
- 缺点：由于类条件独立假设割断了属性间的联系，使得其网络结构不合理，导致了朴素贝叶斯网络分类器的分类精度相对较低。

6.4.2　TAN 贝叶斯网络

Tree Augmented Naive Bayesian(TAN)网络是一种有约束的贝叶斯网络，是对朴素贝叶斯网络分类器的一种改进。它要求属性节点除了以类结构为父节点外最多只能有一个属性父节点，即每一节点至多有两个父节点，如图6.3所示。

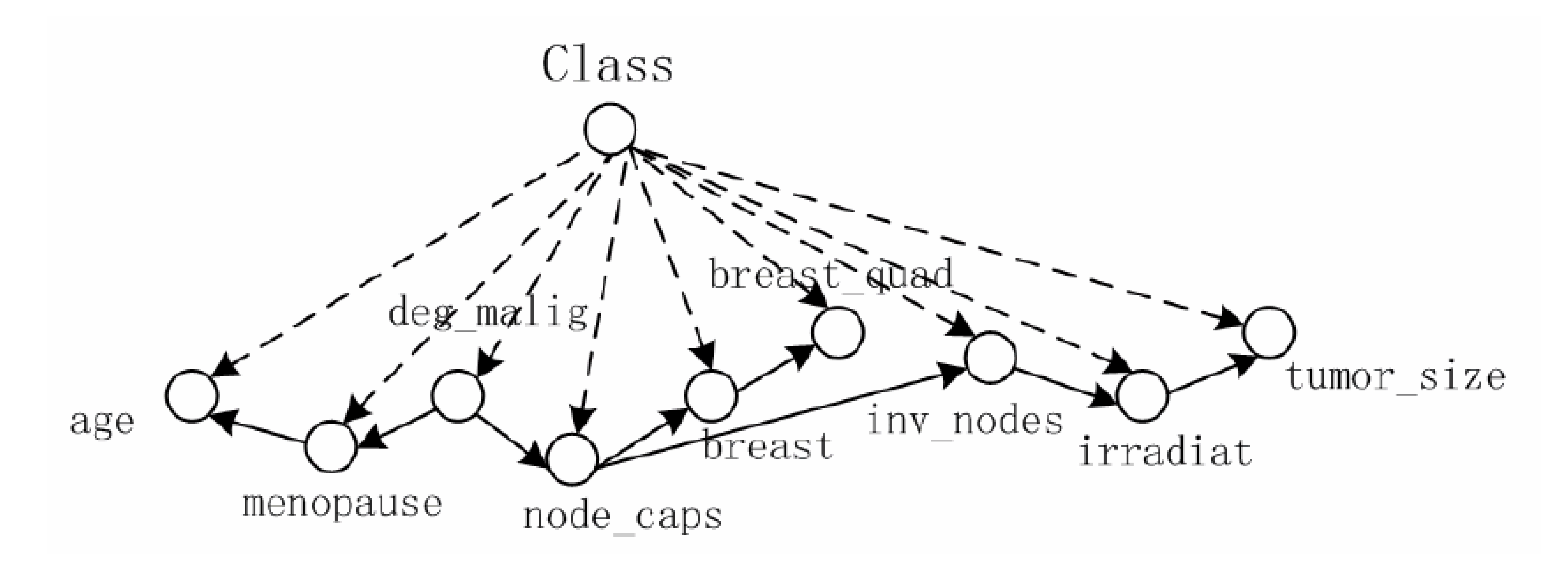


图6. 3　一个TAN网络结构

若X、Y、Z是属性变量，则两变量间的条件互信息定义为

$$I_p(X;Y \mid Z) = \sum_{x,y,z} P(x,y,z) \log_2 \frac{P(x,y \mid z)}{P(x \mid z)P(y \mid z)}$$

它度量一个变量包括另一个变量的信息的多少，两变量间的互信息越大，则两个变量朴素包含对方的信息就越多。

设 $\{X_1, X_2, \cdots, X_n\}$ 是n个属性节点，则TAN的结构学习过程分为如下5个步骤：

（1）计算属性变量对之间的条件信息：$I_p(X_i;X_j|C), i,j=1,2,\cdots,n$

（2）建立一个以 $I_p(X_i;X_j|C)$ 为弧的权重的加权完全无向图，$i,j=1,2,\cdots,n$

（3）找出一个最大权重生成树。

（4）选择一个根节点，设置所有边的方向是由根节点向外，把无向树转换为有向树。

（5）建立一个类变量节点及类变量节点与属性节点之间的弧。

建立最大权重生成树的方法是：首先把边按权重由大到小排序，然后遵照选择的边不能构成回路的原则，按照边的权重由大到小的顺序选择边，这样由所选择的边构成的树便是最大权重生成树。

TAN分类器的特点如下。

网络结构较为简单，建立网络耗时少，由于它在一定程度上克服了朴素贝叶斯网络分类器结构的不合理假设，分类精度较朴素贝叶斯网络高，且其分类性能是当前所有贝叶斯网络分类器中的佼佼者。由于TAN分类器的优异性能以及网络结构简单，TAN分类器是一种被广泛应用的贝叶斯网络分类器。

6.4.3 无约束贝叶斯网络

学习无约束贝叶斯网络结构时需要引入一个评估函数。目前常用的用于学习贝叶斯网络的两个评分函数是贝叶斯评分函数以及基于最小描述长度（MDL）的函数。

设$B=<G,\Theta>$是一贝叶斯网络，$D=\{u_1,u_2,\cdots,u_n\}$是训练样本集，则网络B的评分函数为

$$\mathrm{MDL}(B|D)=\frac{\lg N}{2}|B|-\mathrm{LL}(B|D)$$

其中：$|B|$是网络参数的个数，而LL（$B|D$）为：$\mathrm{LL}(B|D)=\sum_{i=1}^{N}\lg(P_B(u_i))$。

上式给出了已知节点数n时，决定可能的贝叶斯网络结构的个数的回归函数。很明显，随着节点数的增加，相应的可能网络结构个数是呈指数级增长的。因此，当节点数较大时，如何有效地、快速地在其相应的网络结构空间中找出与训练数据匹配最好的网络结构是无约束网络结构学习的重点。

6.5 基于 MATLAB 的贝叶斯网络方法

应用贝叶斯定理进行分类，可以用两种方法。一种是本节介绍的贝叶斯网络方法；另一种方法是基于概率统计的贝叶斯分类器。后者在计算类条件概率密度函数$P(X|C)$时采用以下的公式，此方法适用于连续属性；而对于离散属性要采用贝叶斯网络方法。在应用贝叶斯网络方法处理连续属性时首先要离散化。

在大多数情况下，类条件概率密度可以采用多维变量的正态密度函数来模拟：

$$P(X|\omega_i)=\ln\{\frac{1}{(2\pi)^{n/2}|S_i|^{1/2}}\exp[-\frac{1}{2}(X-\overline{X^{\omega_i}})^T S_i^{-1}(x-\overline{X^{\omega_i}})]\}$$
$$=-\frac{1}{2}(X-\overline{X^{\omega_i}})^T S_i^{-1}(x-\overline{X^{\omega_i}})-\frac{n}{2}\ln 2\pi-\frac{1}{2}\ln|S_i|$$

式中：$X=(x_1,x_2,\cdots,x_n)$为 n 维特征向量；$\overline{\mu}=(\mu_1,\mu_2,\cdots,\mu_n)$ 为 n 维均值向量；$S=E[(X-\overline{\mu})(X-\overline{\mu})^T]$ 为 n 维协方差矩阵；S^{-1} 是 S 的逆矩阵，$|S|$ 是 S 的行列式，$\overline{X}^{\omega_i}$ 为 ω_i 类的均值向量。

例2.43　测定了冠心病人和正常人血中微量元素的含量（如表6.1所示），试用Bayes法进行分类。

表6.1　冠心病人及正常人血中4种微量元素的测定结果（μg/mL）

样 本 号	测定结果				原 归 类
	x_1	x_2	x_3	x_4	
1	0.039	0.980	46.2	6.32	1
2	0.051	0.580	32.9	4.85	1
3	0.009	0.800	50.9	6.48	1
4	0.042	0.920	55.5	6.27	1
5	0.026	1.56	43.2	5.45	1
6	0.034	0.74	59.2	7.13	1
7	0.016	0.75	41.6	4.56	1
8	0.019	0.82	33.2	7.06	1
9	0.037	0.94	36.8	6.21	1
10	0.051	0.87	33.7	6.17	1
11	0.071	1.13	31.4	7.19	1
12	0.055	0.870	35.9	5.53	1
13	0.099	1.100	33.6	7.18	1
14	0.031	0.53	31.9	4.07	2
15	0.030	0.750	53.1	6.48	2
16	0.050	0.790	36.4	4.53	2
17	0.040	0.720	50.0	4.07	2
18	0.043	0.81	65.4	6.18	2
19	0.047	0.640	53.6	4.23	2
20	0.076	0.60	63.5	6.0	2
21	0.072	0.610	44.6	4.49	2
22	0.103	0.75	68.4	7.11	2
23	0.062	0.65	62.1	7.34	2
24	0.087	0.88	70.8	7.78	2
25	0.091	0.73	70.1	6.94	2
26	0.040	0.570	36.7	3.74	2

解：

```
>>load mydata;
>> y=bayes(mydata(1:13,:),mydata(14:26,:),mydata(1:13,:),1);
y =1    1    2    2    1    2    1    1    1    1    1    1    1
```

结果表明，第 4、5、7 号样品分类与原归类不一致。

```
function result=bayes(varargin)  %bayes分类函数
type=varargin{end};             %1为基于最小错误率，2为基于最小风险率
r1=length(varargin)-2; loss=ones(r1)-diag(diag(ones(r1)));
test=varargin{end-1};x=[];
for i=1:r1;x=[x;varargin{i}]; r(i)=size(varargin{i},1);end
[y1,y2]=mypcacov(x,test);       %主成分分析，以保证样本呈正态分布
for k=1:size(test,1)
temp=0;
  for
i=1:r1;y=y1(temp+1:temp+r(i),:);temp=temp+r(i);y_cov=cov(y);y_inv=inv(y_cov);
   y_det=det(y_cov);
if r(i)==1;y_mean=y;else;y_mean=mean(y);end
   p=r(i)/sum(r);h(i)=-(y2(k,:)-y_mean)'*y_inv*(y2(k,:)-y_mean)/2+log(p)-log
(abs(y_det))/2;
end
switch type
case 1
     [a,result(k)]=max(h);      %基于最小错误率的分类
case 2
     for j=1:r1; risk(j)=loss(j,:)*h';end;[a,result(k)]=min(risk);
                                                 %基于最小风险率的分类
end
end
```

例 2.44　利用贝叶斯网络方法对表 6.2 所示的数据进行分类分析。

表6.2　某店顾客情况数据集

RID	age	income	student	credit_rating	Class:buys-computer
1	≤30	high	no	fair	no
2	≤30	high	no	excellent	no
3	31…40	high	no	fair	yes
4	＞40	medium	no	fair	yes
5	＞40	low	yes	fair	yes
6	＞40	low	yes	excellent	no
7	31…40	low	yes	excellent	yes
8	≤30	medium	no	fair	yes

续表

RID	age	income	student	credit_rating	Class:buys-computer
9	≤30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	≤30	medium	yes	excellent	yes
12	31…40	medium	no	excellent	yes
13	31…40	high	yes	fair	yes
14	>40	medium	no	excellent	no

解：

利用朴素贝叶斯网络解此问题，编程如下：

```
>>train ={'age''income' 'student' 'credit_rating' 'Class:buys-computer';
'≤30'  'high' 'no' 'fair' 'no';'≤30' 'high' 'no' 'excellent' 'no';
   '31…40' 'high' 'no' 'fair' 'yes';'>40''medium' 'no' 'fair' 'yes';
   '>40' 'low' 'yes' 'fair' 'yes';'>40' 'low' 'yes' 'excellent' 'no';
   '31…40' 'low' 'yes' 'excellent' 'yes';'≤30''medium' 'no' 'fair' 'no';
   '≤30' 'low' 'yes' 'fair' 'yes';'>40' 'medium' 'yes' 'fair' 'yes';
   '≤30' 'medium' 'yes' 'excellent' 'yes';'31…40' 'medium' 'no' 'excellent' 'yes';
   '31…40'  'high'  'yes'  'fair'  'yes';'>40' 'medium'  'no' 'excellent' 'no'};
>>sample=train(2:6,:);
>> class1=bayesnet(train,sample);   %贝叶斯网络函数
function class1=bayesnet(train,sample)
 [Ni,M]=size(train(2:end,1:end-1));sample_num=size(sample,1);class=unique(tra
in(2:end,end));
 for i=1:M
    property_train{i}.val=unique(train(2:end,i));
    property_train{i}.name=train(1,i);
    property_train{i}.val_num=1/length(property_train{i}.val);
 end
 p_class=zeros(1,length(class));
 for i=1:Ni
    for j=1:length(class)
      if isequal(class(j),train(i+1,end))
         p_class(j)=p_class(j)+1
      end
    end
 end
```

```
 for i=1:M
   p{i}.property=zeros(length(property_train{i}.val),length(class));
   for j=1:length(property_train{i}.val)
     for m=1:length(class)
       a=0;
       for k=1:Ni
         if
isequal(train(k+1,i),property_train{i}.val(j))&&isequal(train(k+1,end),class(m))
            a=a+1;
          end
       end
       p{i}.property(j,m)=a/p_class(m);
       if p{i}.property(j,m)==0;
           p{i}.property(j,m)=(1/Ni)/(p_class(m)+property_train{i}.val_num/Ni);
       end
     end
   end
end
p_class=p_class./Ni;class1=cell(1,sample_num);
for i=1:sample_num
  a=ones(1,length(class));
   for j=1:length(class)
     for k=1:M
       for m=1:length(property_train{k}.val)
         if isequal(sample(i,k),property_train{k}.val(m))
            a(1,j)=a(1,j)*p{k}.property(m,j)
         end
       end
     end
     p1(i,j)=a(1,j)*p_class(j);
   end
  [a,b]=max(p1(i,:)); class1{i}=class(b);
end
```

例 2.45　为分析求解毕业生就业预测问题，给出学生就业数据如表 6.3 所示，其来源于某高校学生就业的情况统计，请用朴素贝叶斯网络法进行分类分析。

表6.3　学生就业数据表

性　别	学生干部	优秀毕业论文	学位获得	综合成绩	就　业
male	no	no	no	90	yes
male	yes	yes	yes	76	no
female	no	no	no	95	yes
male	yes	yes	yes	80	yes
male	no	yes	no	79	no
female	no	no	no	89	no
male	yes	yes	yes	79	yes
female	no	no	no	88	yes
male	no	no	no	86	yes
male	yes	yes	no	75	no
male	yes	no	yes	80	yes
female	yes	yes	yes	90	yes
female	no	yes	yes	95	yes
male	no	no	yes	90	yes
male	no	no	yes	80	no
female	yes	no	yes	80	yes
female	no	no	yes	90	yes
male	no	yes	yes	92	yes
male	no	no	no	72	no
male	no	yes	yes	85	yes

解：

在实际利用朴素贝叶斯网络法进行分类时，可能会遇到以下三种情况：一是属性为连续属性；二是条件概率为零；三是缺少某个属性值。

本例中，其中综合成绩是一个连续属性，此时通常假定该属性服从高斯分布，并按下式计算概率

$$P(x_k \mid c_i) = g(x_k, \mu_{c_i}, \sigma_{c_i}) = \frac{1}{\sigma_{c_i}\sqrt{2\pi}} e^{-\frac{(x-\mu_{c_i})^2}{2\sigma_{c_i}^2}}$$

其中：给定类 C_i 的训练样本属性 x_k 的值；$g(x_k, \mu_{c_i}, \sigma_{c_i})$ 是属性 x_k 的高斯密度函数；μ_{c_i}、σ_{c_i} 分别为平均值和标准差。

当条件概率为零时，可以用下式计算：$P(x_k \mid c_i) = \dfrac{n_c + l \times p}{n + l}$，其中 n 为类 c_i 中的样本常数，n_c 是类 c_i 的训练样集中取值为 x_k 的样例数，l 是称为等价样本大小的参数，而 p 是用户指定的参数，可以看作是在类 c_i 的记录中观察属性值 x_k 的先验概率。决定先验概率和观察概率之间的概率。

对于概率值，即使每个乘积因子都不为零，但当较大时，也可能几乎为零，此时难以区分不同类别。为解决这个问题，可以将乘积问题转化为加法计算问题以避免“溢出”。

$$\log P(x_k \mid c_i) = \log P(c_i) + \sum_{k=1}^{n} \log P(x_k \mid c_i)$$

根据以上的处理方法，可编程计算本例。

```
>> train ={'性别' '学生干部' '优秀毕业论文' '学位获得' '综合成绩' '就业'
'male' 'no' 'no' 'no' 90 'yes';'male' 'yes' 'yes' 'yes' 76 'no';'female' 'no' 'no'
'no' 95 'yes';'male' 'yes' 'yes' 'yes' 80 'yes';'male' 'no' 'yes' 'no' 79
'no';'female' 'no' 'no' 'no' 89 'no';'male' 'yes' yes' 'yes' 79 'yes';'female'
'no' 'no' 'no' 88 'yes';'male' 'no' 'no' 'no' 86 'yes'; 'male' 'yes' 'yes' 'no'
75 'no' 'male' 'yes' 'no' 'yes' 80 'yes' 'female' 'yes' 'yes' 'yes' 90
'yes';'female' 'no' 'yes' 'yes' 95 'yes' 'male' 'no' 'no' 'yes' 90 'yes';'male'
'no' 'no' 'yes' 80 'no';female' 'yes' 'no' 'yes' 80 'yes';'female' 'no' 'no' 'yes'
90 'yes';'male' 'no' 'yes' 'yes' 92 'yes';'male' 'no' 'no' 'no' 72 'no';'male'
'no' 'yes' 'yes' 85 'yes'};
sample={'male' 'yes' 'no' 'no' 82;'female' ' ' 'yes' 'no' 88};
numeric=[5];
class1=bayesnet1(train,sample,numeric);
>>class1{1}='no'  class1{2}='yes'
```

例 2.46 利用 TAN 贝叶斯网络对例 2.43 的数据进行分类分析。

解：

首先确定 TAN 贝叶斯网络结构，其方法：一种是首先给变量排序，然后确定变量之间的条件独立性，再根据变量的关系确定网络结构；另一种方法（更为常用）是用户根据变量之间的因果关系（根据用户的已有知识）来建立网络结构。

然后再构造 TAN 分类器，其方法：一种是由 Friedman 等人提出的基于分布的构造方法；另一种是由 Eamonn 和 Pazzani 提出的基于分类的构造方法。基于分类的方法的分类性能比基于分布的方法的分类性能更优，但是，由于每条增强弧的选择都需要评估函数的评测，所需的构造时间比基于分布的方法所需的时间长得多。

根据以上原理，可编程对问题进行分析，其中 TAN 结构为图 6.4 所示。

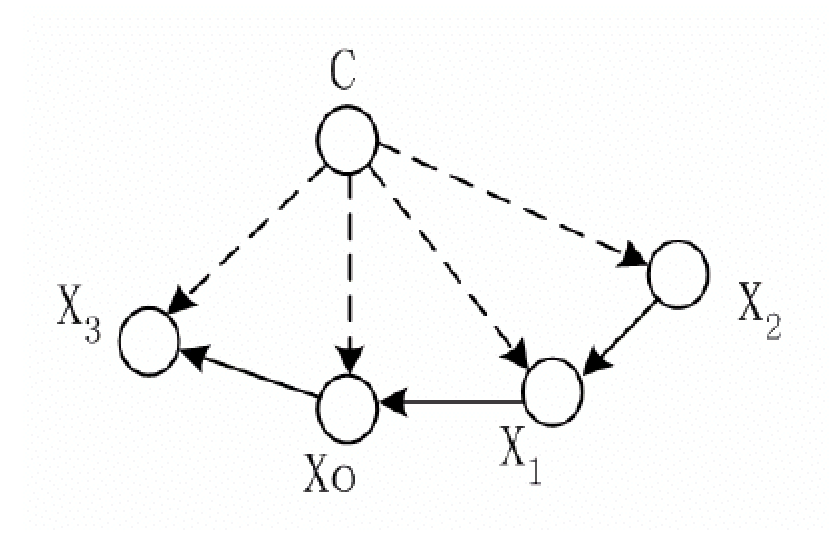


图6.4　TAN结构

```
>>train ={'age''income' 'student' 'credit_rating' 'Class:buys-computer';
'≤30'  'high' 'no' 'fair' 'no';'≤30' 'high' 'no' 'excellent' 'no';
   '31…40' 'high' 'no' 'fair' 'yes';'>40''medium' 'no' 'fair' 'yes'
   '>40' 'low' 'yes' 'fair' 'yes';'>40' 'low' 'yes' 'excellent' 'no';
   '31…40' 'low' 'yes' 'excellent' 'yes';'≤30''medium' 'no' 'fair' 'no';
   '≤30' 'low' 'yes' 'fair' 'yes';'>40' 'medium' 'yes' 'fair' 'yes';
   '≤30' 'medium' 'yes' 'excellent' 'yes';'31…40' 'medium' 'no' 'excellent' 'yes';
   '31…40'  'high'  'yes'  'fair'  'yes';'>40' 'medium'  'no' 'excellent' 'no'};
   sample={'≤30'     'medium'    'yes'        'fair'};
   str(1).net=2;str(2).net=3;str(3).net=[];str(4).net=1;
>>class=bayesTAN(train,sample,str)
>>class{1}='yes'
```

例 2.47　应用无约束贝叶斯网络方法对例 2.43 的数据进行分类分析。

解：

根据无约束贝叶斯网络方法的原理，构造如图 6.5 所示的无约束贝叶斯网络结构，并据此编程，得到如下的结果。此结果与前例有所差异，这是由本例中各节点的条件概率计算方法与前不同所引起的。

```
>>load mydata;sample={'≤30' 'medium' 'yes' 'fair'};
>>str(1).net=[2 4 5];str(2).net=[];str(3).net=2;str(4).net=[];str(5).net=3;
>> class1=bayesTAN1(train,sample,str);
>> class1='no'
```

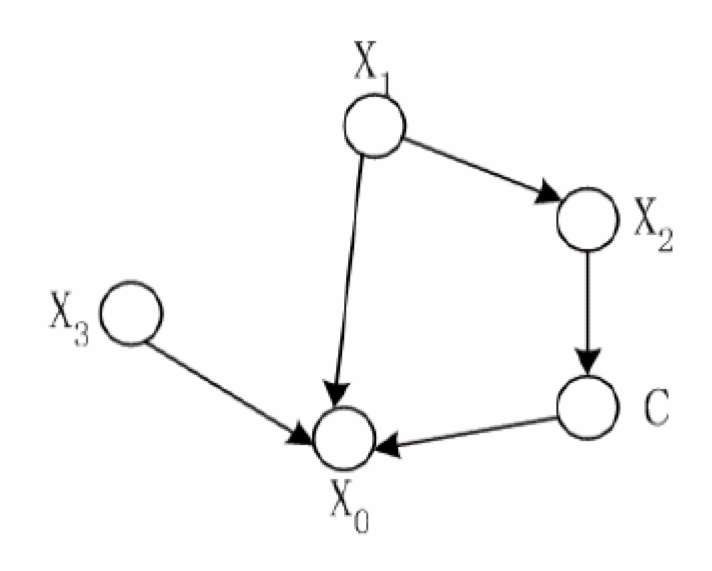


图6.5　无约束贝叶斯网络结构

例 2.48　对于贝叶斯判别，MATLAB 中提供了 NaiveBayes 类，可以根据训练集创建一个类对象，一个类对象定义了一个朴素贝叶斯判别分类器，利用这个分类器便可以对未知类别的样本进行分类。

试利用 NaiveBayes 类函数对某葡萄酒数据库进行分析，数据库为 178×14 矩阵，其中前两列为类别号，后 13 列为决定酒质量的 13 个属性，包括 Alcohol、Malic acid、 Ash、Alcalinity of ash、Magnesium、Total phenols、Flavanoids、Nonflavanoid phenols、Proanthocyanins、Color intensity、Hue、OD280/OD315 of diluted wines 和 Proline 等指标。

解：

```
>> a=dlmread('D:\wine.txt');class=a(:,1);
>> wine1=a(1:59,2:end);wine2=a(60:130,2:end);wine3=a(131:178,2:end);
>> pre1=obj.predict(wine1);
>> obj=NaiveBayes.fit(a(:,2:end),class);      %创建一个 NaiveBayes 分类器
>> pre1=obj.predict(wine1);                    %对第 1 类酒的样本进行预测
```

从结果看，样本全部分类正确。

例 2.49　朴素贝叶斯分类假定类条件独立，即给定样本的类标号，属性的值相互条件独立。但在实践中变量之间的依赖可能存在。

贝叶斯信念网络说明联合条件概率分布，它允许在变量的子集间定义类条件独立性，它提供一种因果关系的图形。

假设某服装零售商在城市 A 和 B 各开设一家服装店（店 1 和店 2），均四季销售。该零售商对三种服装尤为关注：某品牌大衣、某品牌衬衫和某品牌裤子，关注内容包括服装面料的重量（轻、中等、重）和颜色（暖色、中度、冷色）。

表 6.4 为这三种服装的销售情况统计表；表 6.5 为服装的相关数据统计表。请对此数据集进行分析，以帮助零售商做出决策。

表6.4　服装销售情况统计表（件）

季节 \ 商店	店 1			店 2		
	大衣	衬衫	裤子	大衣	衬衫	裤子
春季	30	30	40	30	60	60
夏季	10	30	60	8	52	90
秋季	40	40	20	30	75	45
冬季	60	35	5	45	60	45

表6.5　100件服装面料重量和颜色情况统计表（件）

类别 \ 指标	面料重量			面料颜色		
	轻	中等	重	暖色	中度	冷色
大衣	10	20	70	10	30	60
衬衫	20	60	20	70	20	10
裤子	50	40	10	30	40	30

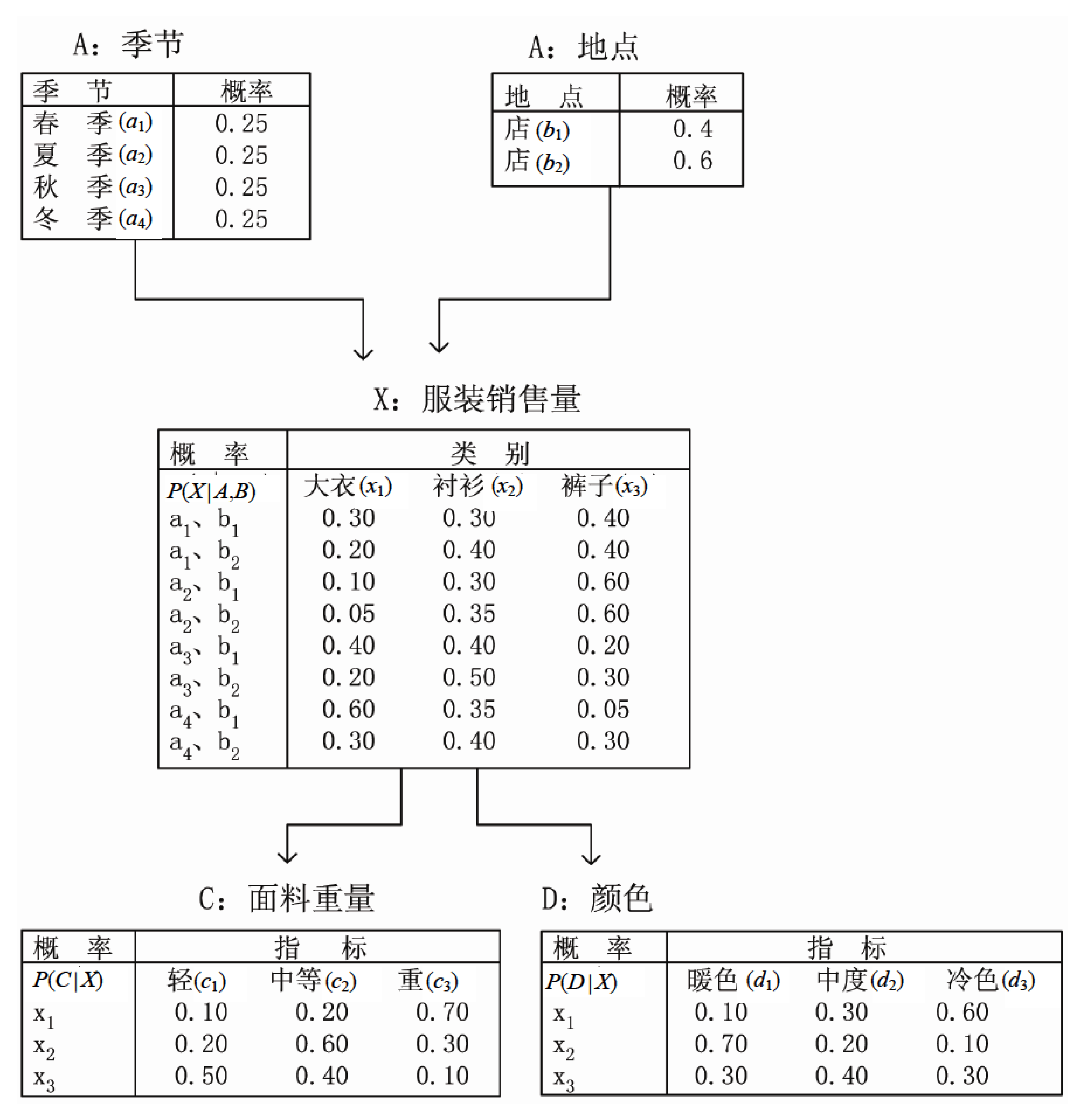
解：

建立一个贝叶斯网络，它需考虑两个方面因素：变量间的关系和相关的局部概率。

该零售商共有 5 个变量：季节、地点、服装购买量、面料重量和颜色。在这些变量中，季节和其他变量均无关系（冬天购买衬衫并不代表夏天来了）。因此将代表季节变量的节点置于贝叶斯网络的顶端，也即意味着该变量与其他变量不存在相关关系。

同样，地点也与其他变量无关，因此也置于贝叶斯网络的顶端。面料的重量和颜色要到购买的时候才能知道，因此代表服装购买量的变量节点被插在贝叶斯网络中，且弧线分别指向面料重量节点和颜色节点。

建立贝叶斯网络的第二点是明确每个节点的概率表中的数据。根据表 2.42 和表 2.43 中的数据可以得到图 6.6 所示的贝叶斯网络图。



季　节	概率
春　季 (a_1)	0.25
夏　季 (a_2)	0.25
秋　季 (a_3)	0.25
冬　季 (a_4)	0.25

地　点	概率
店 (b_1)	0.4
店 (b_2)	0.6

概　率	类　别		
$P(X\|A,B)$	大衣 (x_1)	衬衫 (x_2)	裤子 (x_3)
a_1、b_1	0.30	0.30	0.40
a_1、b_2	0.20	0.40	0.40
a_2、b_1	0.10	0.30	0.60
a_2、b_2	0.05	0.35	0.60
a_3、b_1	0.40	0.40	0.20
a_3、b_2	0.20	0.50	0.30
a_4、b_1	0.60	0.35	0.05
a_4、b_2	0.30	0.40	0.30

概　率	指　标		
$P(C\|X)$	轻 (c_1)	中等 (c_2)	重 (c_3)
x_1	0.10	0.20	0.70
x_2	0.20	0.60	0.30
x_3	0.50	0.40	0.10

概　率	指　标		
$P(D\|X)$	暖色 (d_1)	中度 (d_2)	冷色 (d_3)
x_1	0.10	0.30	0.60
x_2	0.70	0.20	0.10
x_3	0.30	0.40	0.30

图 6.6　贝叶斯网络图

根据图 6.6 所示的贝叶斯网络图就可以计算各种情况下的概率。例如计算店 2 冬季的面料为轻、颜色为中度的裤子的购买概率：

$$p(A=a_4, B=b_1, C=c_1, D=d_2, X=x_3)$$
$$= p(\text{季节=冬季}) \times p(\text{地点=店1}) \times p(\text{服装=裤子} \mid \text{季节=冬季且地点=店1})$$
$$\times p \times (\text{面料=轻} \mid \text{服装=裤子})) \times p(\text{颜色=中度} \mid \text{服装=裤子})$$
$$= p(A=a_4) \times p(B=b_1) \times p(X=x_3 \mid A=a_4 \quad B=b_1) \times p(C=c_1 \mid X=x_3)$$
$$p \times (D=d_2 \mid X=x_3) \times p(D=d_2 \mid X=x_3)$$
$$=0.25 \times 0.4 \times 0.05 \times 0.50 \times 0.40 = 0.001$$

显然，在店 1 所在城市 1，冬天对于面料重量为轻、色彩为中度的裤子的需求量不大。用同样的方式可以计算出任何季节、地点、服装种类、面料重量和颜色各种组合的概率。

在贝叶斯网络上还可以计算出每个节点的先验概率。例如一件大衣的先验概率如下：

$$
\begin{aligned}
p(\text{大衣}) = p(X = x_1) \\
&= p(X = x_1 \mid A = a_1 \cap B = b_1)p(A = a_1 \cap B = b_1) \\
&+ p(X = x_1 \mid A = a_1 \cap B = b_2)p(A = a_1 \cap B = b_2) \\
&+ p(X = x_1 \mid A = a_2 \cap B = b_1)p(A = a_2 \cap B = b_1) \\
&+ p(X = x_1 \mid A = a_2 \cap B = b_2)p(A = a_2 \cap B = b_2) \\
&+ p(X = x_1 \mid A = a_3 \cap B = b_1)p(A = a_3 \quad B = b_1) \\
&+ p(X = x_1 \mid A = a_3 \cap B = b_2)p(A = a_3 \cap B = b_2) \\
&+ p(X = x_1 \mid A = a_4 \cap B = b_1)p(A = a_4 \cap B = b_1) \\
&+ p(X = x_1 \mid A = a_4 \quad B = b_2)p(A = a_4 \quad B = b_2) \\
&= 0.30 \times 0.10 + 0.20 \times 0.15 + 0.10 \times 0.10 + 0.05 \times 0.15 \\
&+ 0.40 \times 0.10 + 0.20 \times 0.15 + 0.60 \times 0.10 + 0.30 \times 0.15 = 0.2525
\end{aligned}
$$

计算中已经假定季节和地点是相互独立的，因此：$p(\mathrm{A} \cap \mathrm{B}) = p(A)p(B)$ 也可以计算后验概率。例如：

$$
p(\text{冬节} \mid \text{大衣}) = \frac{p(\text{大衣}}{p(\text{冬节} \cap \text{大衣})}
$$

而：

$$
\begin{aligned}
p(\text{冬节} \cap \text{大衣}) &= p(\text{冬节} \cap \text{店1} \cap \text{大衣}) + p(\text{冬节} \cap \text{店2} \cap \text{大衣}) \\
&= p(\text{冬节}) \ p(\text{店1}) \ p(\text{大衣} \mid \text{季节} \cap \text{店1}) \\
&+ p(\text{冬节}) \ p(\text{店2}) \ p(\text{大衣} \mid \text{季节} \cap \text{店2}) \\
&= 0.25 \times 0.4 \times 0.6 + 0.25 \times 0.6 \times 0.3 = 0.105
\end{aligned}
$$

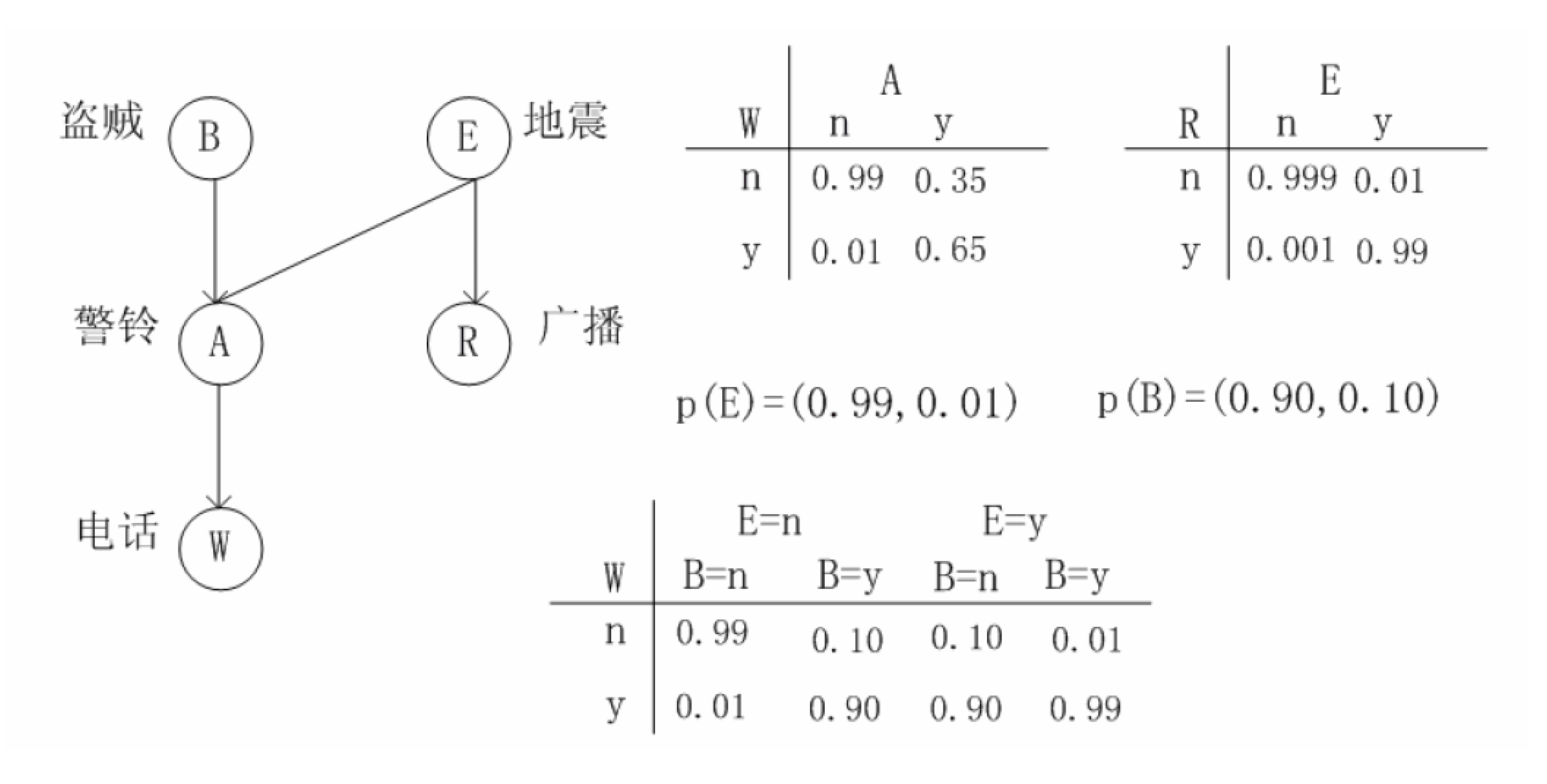
则：

$$
p(\text{冬节} \mid \text{大衣}) = \frac{p(\text{大衣})}{p(\text{冬节} \cap \text{大衣})} = \frac{0.2525}{0.105} = 0.4158
$$

这样，贝叶斯网络就能根据 $p(\text{冬季} \mid \text{大衣})$、$p(\text{春季} \mid \text{大衣})$、$p(\text{夏季} \mid \text{大衣})$和 $p(\text{秋季} \mid \text{大衣})$之间的最大后验概率做出一个季节选择的决定，以确定各商店的销售策略。

例 2.50　福尔摩斯先生在他的办公室工作时接到了他邻居华生的电话。华生告诉他：他的家里可能进了窃贼，因为他家的警铃响了。

被告知有窃贼闯入，福尔摩斯迅速开车回家。在路上，他听广播得知他家那里发生了地震。地震也有可能引起警报。这样，请问福尔摩斯先生应该回家抓贼还是迅速撤离该地区以躲避地震？图 6.7 为计算所需的各概率。



W	A=n	A=y
n	0.99	0.35
y	0.01	0.65

R	E=n	E=y
n	0.999	0.01
y	0.001	0.99

W	E=n B=n	E=n B=y	E=y B=n	E=y B=y
n	0.99	0.10	0.10	0.01
y	0.01	0.90	0.90	0.99

图6.7　网络结构图及各类概率值

解：

利用网上下载的贝叶斯网络工具箱 FullBNT-1.0.4 求解本问题。

在路上的福尔摩斯需要判断是盗贼还是地震导致警铃？如果是前者，他需要回去抓贼，若是后者，则要逃离地震区。

所以图中虽然有 5 个节点，地震并不 100%导致警铃，警铃也不 100%导致的电话。

但是我们在得到电话听到警铃的情况下，可以通过计算盗贼导致警铃的概率 p1，和地震导致警铃的概率 pp1 来进行决策，也可以计算在地震发生条件下，盗贼导致警铃的概率 p2。如果 p2 比 p1 小，说明新添加的条件 E 才是导致 A 的主要原因。

```
N = 3;                                          %三个节点，分别是 B、E、A
dag = zeros(N,N);B = 1; E = 2; A = 3;
dag(B,A) = 1;dag(E,A) = 1;                      %节点之间的连接关系
discrete_nodes = 1:N;                           %离散节点
node_sizes = 2*ones(1,N);                       %节点状态数
bnet =mk_bnet(dag,node_sizes,'names',{ 'BB','EE','AAA'},'discrete',discrete_nodes);
bnet.CPD{B} = tabular_CPD(bnet,B,[0.9 0.1]);   %手动输入的条件概率
bnet.CPD{E} = tabular_CPD(bnet,E,[0.99 0.01]);
bnet.CPD{A} = tabular_CPD(bnet,A,[0.99 0.1 0.1 0.01 0.01 0.9 0.9 0.99]);
draw_graph(dag);                                %画贝叶斯结构图如图 6.8 所示
engine = jtree_inf_engine(bnet);          %使用联合树引擎对贝叶斯网络进行推断
%求解边缘分布假设，计算盗窃导致响铃的概率
evidence = cell(1,N);evidence{A} = 2;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, B);
```

```
p1 = marg.T(2);
p1=0.8412                                    %盗贼导致警铃的概率
%现在添加地震的证据观察它有什么不同
evidence{E} = 2;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, B);
p2 = marg.T(2);
2=0.1089                                     %结论是地震更能解释响铃这个主要事实
```

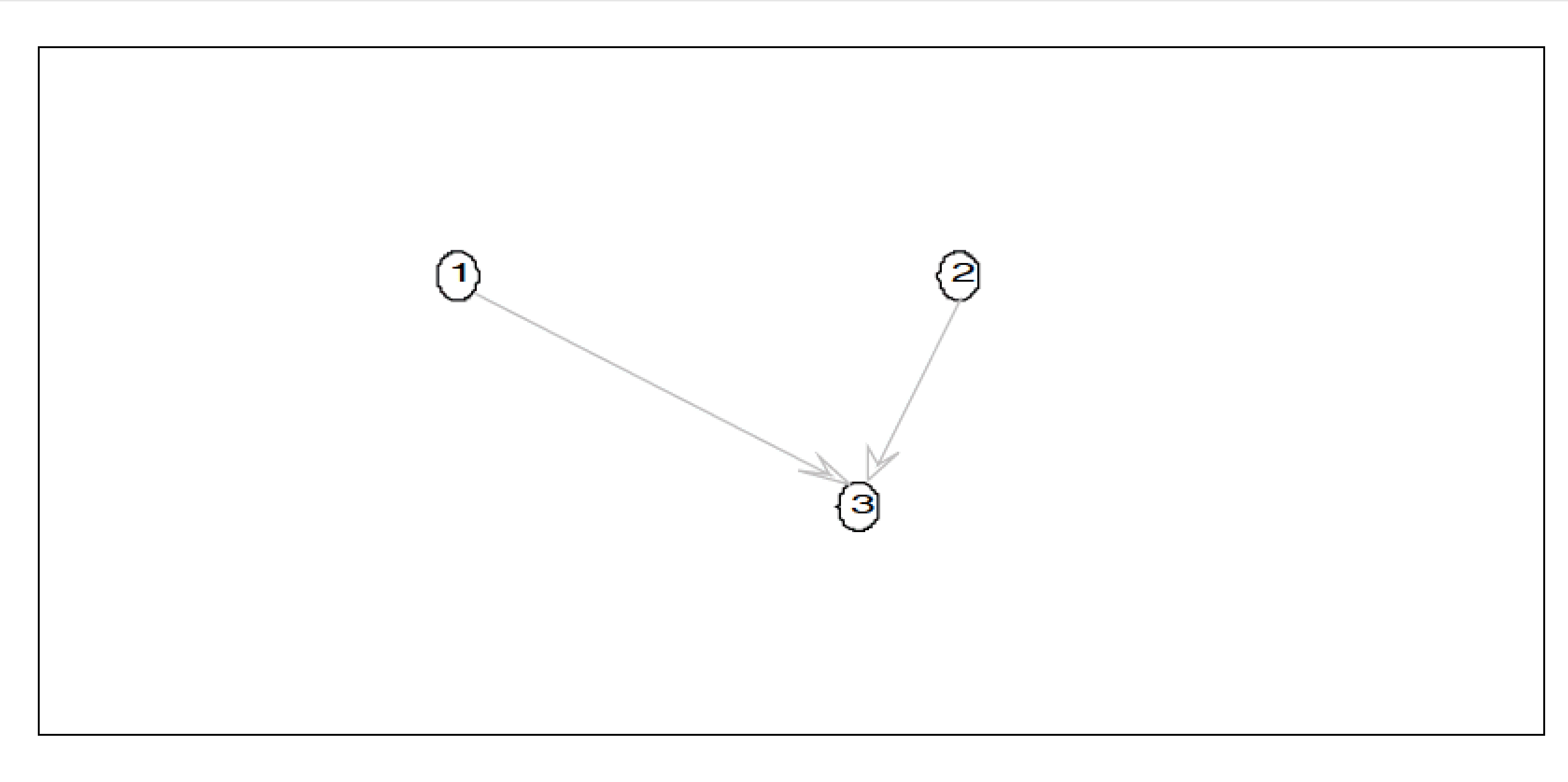


图6.8　三节点贝叶斯网络结构图

现在添加 R 和 W 两个节点，再进行相关的计算。

```
N = 5; %三个节点，分别是 B、E、A、R、W
dag = zeros(N,N);B = 1; E = 2; A = 3; R = 4; W=5;
dag(B,A) = 1;dag(E,A) = 1; dag(E,R) = 1;dag(A,W) = 1; %节点之间的连接关系
discrete_nodes = 1:N;                                 %离散节点
node_sizes = 2*ones(1,N);                             %节点状态数
bnet =mk_bnet(dag,node_sizes,'names',{ ' BB','EE','AAA','RR','WWW'},'discrete
',discrete_nodes);
bnet.CPD{B} = tabular_CPD(bnet,B,[0.9 0.1]);          %手动输入的条件概率
bnet.CPD{E} = tabular_CPD(bnet,E,[0.99 0.01]);
bnet.CPD{A} = tabular_CPD(bnet,A,[0.99 0.1 0.1 0.01 0.01 0.9 0.9 0.99]);
bnet.CPD{R} = tabular_CPD(bnet,R,[0.999 0.01 0.001 0.99]);
bnet.CPD{W} = tabular_CPD(bnet,W,[0.99 0.35 0.01 0.65]);
draw_graph(dag);                            %画贝叶斯结构图如图 6.9 所示
engine = jtree_inf_engine(bnet);            %使用联合树引擎对贝叶斯网络进行推断
%求解边缘分布假设，计算盗窃导致响铃的概率
evidence = cell(1,N);evidence{A} = 2;
```

```
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, B);
p1 = marg.T(2);
p1=0.8412                                   %盗贼导致警铃的概率
%现在添加地震的证据观察它有什么不同
evidence{E} = 2;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, B);
p2 = marg.T(2);
p2=0.1089%                                  %结论是地震更能解释响铃这个主要事实
```

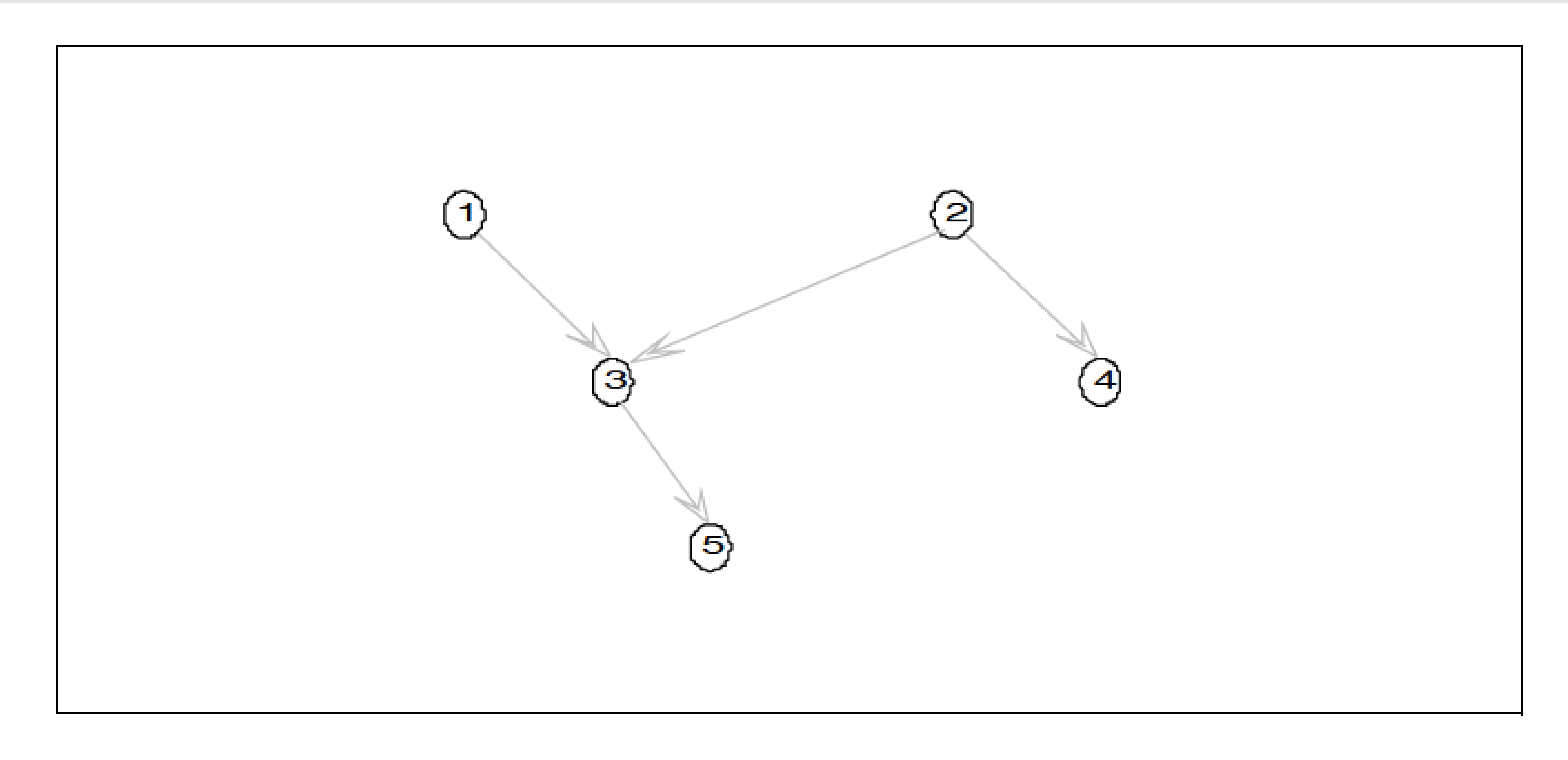


图6.9　五节点贝叶斯网络结构图

从运行结果可看出 R、W 节点对决策无影响。

```
%联合概率分布
evidence = cell(1,N);
[engine, ll] = enter_evidence(engine, evidence);
m = marginal_nodes(engine, [B E A]);
>> m.T
ans(:,:,1) =0.8821    0.0009
        0.0099    0.0000
ans(:,:,2) =0.0089    0.0081
        0.0891    0.0010
```

读书笔记

第 7 章

支持向量机

7.1 支持向量机概述

传统的统计研究方法都是建立在大数定理这一基础上的渐进理论，要求学习样本数目足够多。然而在实际应用中，由于各个方面的原因，这一前提往往得不到保证。因此在小样本的情况下，建立在传统统计学基础上的机器学习方法，也就很难取得理想的学习效果和泛化性能。

针对小样本问题，以 Bell 实验室 V.Vapnik 教授为首的研究小组从 20 个世纪 60 年代开始，就致力于这个问题的研究，并提出了统计学习理论（Statistical Learning Theory，SLT）。支持向量机（Support Vector Machine，SVM）即是统计学习理论发展的产物。针对有限样本情况，SVM 建立了一套完整的、规范的基于统计的机器学习理论和方法，大大减少了算法设计的随意性，克服了传统统计学中经验风险与期望风险可能具有较大差别的不足。目前，SLT 和 SVM 已成为继人工神经网络以来机器学习领域中的研究热点，在模式识别、函数逼迫、概率密度估计、降维等方面获得越来越广泛的应用。

与神经网络相比，SVM 有坚实的统计学基础，它具有以下优点。

（1）以结构风险最小原理为基础，减少推广错误的上界，具有很好的推广性能，解决了神经网络的过拟合问题。

（2）问题的求解等价于线性约束的凸二次规划问题，具有全局最优解，解决了神经网络的局部极小问题。

（3）把原问题映射到高维空间，通过在高维空间构造线性分类函数来实现原问题的划分，引入核函数，解决了维数灾难问题。

对于两类线性可分问题，如图 7.1 所示。分割线（平面 1）1 和分割线（平面 2）2 都能正确地将两类样本分开，即都能保证使经验风险最小（为 0），这样的线（平面）有无限多个，但分割线 1 离两类样本的间隙最大，称为最优分类线（平面）。最优分类线（平面）的置信范围最小。

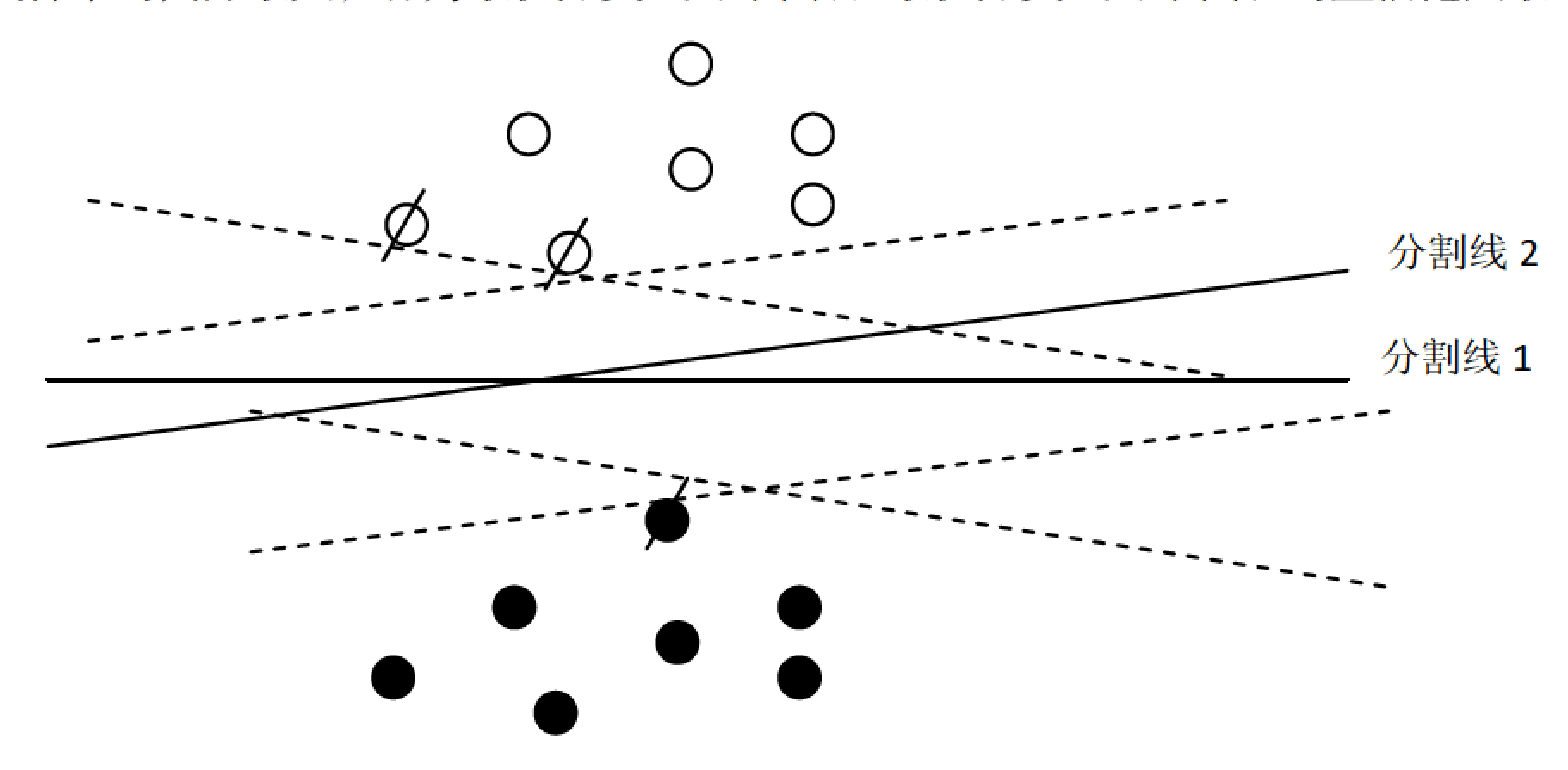


图 7.1　支持向量机原理示意图

设线性可分样本集为（X_i，y_i）（i=1,2,⋯,n；$X\in R^d$，$y\in\{-1,1\}$是类别标号）。d 维空间中线性判别函数的一般形式为 $g(X)=W\cdot X+b$，分类面方程为

$$W\cdot X+b=0$$

将判别函数归一化，然后等比例调节系数 W 和 b，使两类所有样本都能满足$|g(X)|\geqslant 1$，这时

分类间隔为 $2/\|W\|$。这样将求间隔最大变为求 $\|W\|$ 最小。

满足 $|g(X)|=1$ 的样本点，离分类线（平面）距离最小。它们决定了最优分类线（平面），称为支持向量机（support vectors，SV），图中带斜线的 3 个样本即为 SV。

可见，求最优分类面的问题转化为优化问题

$$\min \phi(W)=\frac{1}{2}\|w\|^2=\frac{1}{2}(W \bullet W)$$

$$\text{s.t.}\quad y_i[(w \bullet X_i)+b]-1 \geqslant 0 \quad (i=1,2,\cdots,n)$$

本优化问题可转化为对偶化问题

$$\min Q(\alpha)=\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j (X_i \bullet X_j)-\sum_{i=1}^{n}\alpha_i$$

$$\text{s.t.}\quad \alpha_i \geqslant 0 \quad (i=1,2,\cdots,n)$$

$$\sum_{i=1}^{n} y_i\alpha_i = 0$$

为叙述和求解的方便，将上式改写成矩阵形式

$$\min Q(\alpha)=\frac{1}{2}\alpha^T A\alpha - b^T\alpha$$

$$\text{s.t.}\quad \alpha_i \geqslant 0 \quad (i=1,2,\cdots,n)$$

$$y^T\alpha = 0$$

式中：$\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_n)^{\mathrm{T}}$，$b=(1,1,\cdots,1)^{\mathrm{T}}$，$y=(y_1,y_2,\cdots,y_n)$，$A_{ij}=y_iy_j(x_i \bullet x_j)$。

由此可得到最优分类函数为

$$f(x)=\operatorname{sgn}\{\sum_{i=1}^{n}\alpha_i^* y_i (X_i \bullet X)+b^*\}$$

因为对于非支持向量满足 $\alpha_i=0$，所以最优函数只需对支持向量进行，而 b^* 可根据任何一个支持向量的约束条件求出。

对于非线性可分问题，可以把样本 x 映射到某个高维空间中去，然后在高维空间中，使用上面的方法。设该映射为 $\boldsymbol{\Phi}$

$$x \to \Phi(X)=\begin{pmatrix}\phi_1(x)\\ \vdots \\ \phi_i(x) \\ \vdots\end{pmatrix}$$

上述的对偶问题变为

$$\min Q(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \Phi(X_i)\Phi(X_j)$$

分类函数为

$$d(x)=w^*\Phi(x)+b^*=\sum_{i,j=1}^{n}\alpha_i y_i\Phi(x)\Phi(x_i)+b^*$$

可以看出，它只涉及样本变换高维空间的内积，而这种内积可以用原空间的函数来实现。

7.2 核函数

对于线性不可分问题，有两种解决途径，一是一般线性化方法，引入松弛变量，此时的优化问题为

$$\min\phi(W)=\frac{1}{2}\|w\|^2=\frac{1}{2}(W\bullet W)+C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.}\quad y_i[(w\bullet X_i)+b]-1+\xi_i\geqslant 0\qquad(i=1,2,\cdots,n)$$

二是 V.Vapnik 引入的核空间理论：将低维输入空间中的数据通过非线性函数映射到高维属性空间 H（也称为特征空间），将分类问题转化到属性空间进行。可以证明，如果选用适当的映射函数，输入空间线性不可分问题在属性空间将转化成线性可分问题。

因此，如果能找到一个映射函数 k 使得 $k(X_i,X_j)=\langle\Phi(X_i),\Phi(X_j)\rangle$，这样在高维特征空间中实际上只需进行内积运算，而这种内积运算可以用输入空间中的某些特殊函数来实现，甚至没有必要知道变换的具体过程。这种特殊的函数称为核函数。根据泛函的有关理论，只要核函数满足 Mercer 条件，它就对应某一变换空间中的内积。

Mercer 定理将核解释为特征空间的内积，它将低维向高维映射，却不需要过多地考虑维数对学习机器性能的影响。核函数是支持向量机的重要组成部分。根据 Hilbert-Schmidt 定理，只要变换 Φ 满足 Mercy 条件，就可以构建核函数。Mercy 条件如下：给定对称函数 $k(x,y)$和任意函数 $\varphi(x)\neq0$，满足约束

$$\int_{-\infty}^{+\infty}\varphi^2(x)\mathrm{d}x<0$$

$$\int\int_{-\infty}^{+\infty}k(x,y)\varphi(x)\varphi(y)\mathrm{d}x\mathrm{d}y>0$$

引入核函数后，以上各式中向量的内积都可用核函数代替

$$\min Q(\alpha)=\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(X_i,X_j)-\sum_{i=1}^{n}\alpha_i$$

$$\text{s.t.}\quad \alpha_i\geqslant 0\qquad(i=1,2,\cdots,n)$$

$$\sum_{i=1}^{n}y_i\alpha_i=0$$

相应的分类函数变为

$$f(x)=\mathrm{sgn}\{\sum_{i=1}^{n}\alpha_i^* y_i K(X_i,X)+b^*\}$$

任选一支持向量，可从下式求出 b^*

$$y_i[\sum_{i=1}^{n}\alpha_i^* y_i K(X_i,X)+b^*]=1$$

目前使用的核函数主要有 4 种：线性核函数、p 阶多项式核函数、多层感知器核函数和 RBF 核函数。

（1）线性核函数：$k(X,Y)=\langle X\cdot Y\rangle$

（2）多项式核函数：$k(X,Y)=\langle\langle X\cdot Y\rangle+c\rangle^p$，其中 c 为常数、p 为多项式阶数，当 c=0，p=1 时即为线性核函数。

（3）多层感知器核函数（Sigmoid）：$k(X,Y)=\tanh(scale\times\langle X\cdot Y\rangle-offset)$，其中 $scale$ 和 $offset$ 是尺度和衰减参数。

（4）RBF 核函数：$k(X,Y)=\exp\left\{\frac{|X-Y|^2}{2\sigma^2}\right\}$，其中 $|\boldsymbol{X}-\boldsymbol{Y}|$ 为两个向量之间的距离，σ 为常数。

从上述的讨论可以看出，应用 SVM 进行分类的步骤为：

① 选择合适的核函数。

② 求解优化方程，获得支持向量及相应的 Lagrange 算子。

③ 写出最优分界面的方程。

④ 根据 sgn($f(X)$)的值，输出类别。

图 7.2 为 SVM 的结构示意图。支持向量机利用输入空间的核函数取代了高维特征空间中的内积运算，解决了算法可能导致的“维数灾难”问题：在构造判别函数时，不是对输入空间的样本作非线性变换，然后在特征空间中求解，而是先在输入空间比较向量，对结果再作非线性变换。这样大的工作量将在输入空间而不是在高维特征空间中完成。

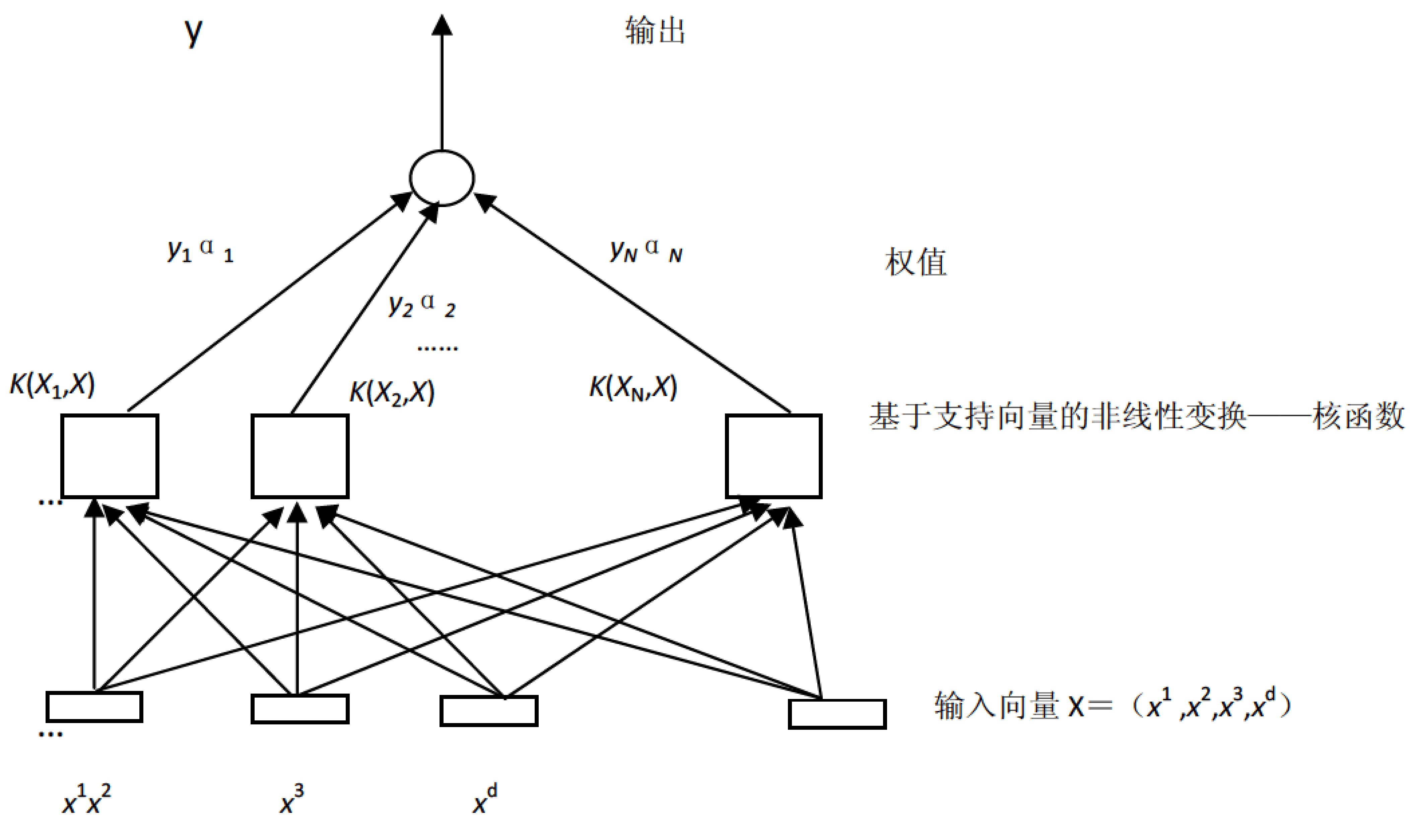


图 7.2　SVM 的结构示意图

7.3 基于 MATLAB 的支持向量机方法

基于 MATLAB 的支持向量机在解决实际问题时，既可以根据支持向量机的原理自己编程，也可以利用 MATLAB 中的自带函数或者各种支持向量机软件包。

例 2.51　试用支持向量机，对表 7.1 中的企业家综合素质作出更为有效的评价，其中 I_i 为各项指标。

解：

在 MATLAB 中，向量机的训练、分类函数分别是 svmtrain 和 svmclassify。一个向量机只能解决二类分类问题，而本例是一个三类分类问题且每类样本数较少，所以用三个分类器，核函数为较为简单的一阶多项式。

```
>>load
mydata;high=mydata(1:3,:);mid=mydata(4:6,:);low=mydata(7:9,:);test=mydata(10:
11,:);
>>num=nchoosek(1:3,2);   %1,2,3三个数字两两配对，1代表高，2代表中，3代表低
>>Training={high,mid,low};SVM=cell(size(num,1),1);      %元胞形式的训练集及SVM
>>for k=1:size(num,1)
   t1=Training{num(k,1)}; t2=Training{num(k,2)};        %配对组成训练集
    SVM{k}=svmtrain([t1;t2],[ones(size(t1,1),1);zeros(size(t2,1),1)],'Kernel_
function',…
    'polynomial','polyorder',1);                        %训练函数
  end
>>for kk=1:size(test,1)
  for k=1:length(SVM)
    result(k)=svmclassify(SVM{k},test(kk,:));           %分类函数
    temp(k)=num(k,1).*result(k)+num(k,2).*~result(k);  %每个分类器的分类结果
  end
  results(kk)=mode(temp,2);              %依据每个数字出现的次数，决定总的分类结果
  end
  >>results
    results =2     2                                    %即都为中等素质
```

表 7.1　企业家素质评价指标数据

评价	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}
高	0.8	0.8	0.9	0.7	0.8	0.7	0.8	0.8	0.8	0.7	0.8	0.7	0.9	0.8	0.7	0.8	0.6
	0.8	0.9	0.7	0.8	0.9	0.8	0.8	0.8	0.8	0.8	0.8	0.7	0.8	0.7	0.6	0.8	0.8
	0.8	0.8	0.9	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8

续表

评价	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}
中	0.7	0.7	0.6	0.7	0.8	0.7	0.6	0.8	0.7	0.6	0.7	0.7	0.6	0.8	0.7	0.7	0.7
	0.7	0.7	0.6	0.6	0.7	0.6	0.7	0.7	0.7	0.7	0.8	0.7	0.6	0.7	0.8	0.7	0.8
	0.7	0.6	0.8	0.7	0.6	0.6	0.7	0.8	0.6	0.7	0.7	0.8	0.8	0.7	0.6	0.7	0.8
低	0.4	0.5	0.5	0.5	0.6	0.5	0.5	0.5	0.5	0.5	0.6	0.5	0.5	0.6	0.7	0.6	0.6
	0.5	0.5	0.5	0.5	0.7	0.6	0.5	0.4	0.5	0.5	0.6	0.5	0.5	0.6	0.5	0.6	0.5
	0.5	0.6	0.5	0.6	0.6	0.5	0.4	0.5	0.5	0.5	0.4	0.5	0.5	0.5	0.7	0.5	0.6
未知	0.8	0.7	0.6	0.9	0.7	0.6	0.8	0.6	0.6	0.7	0.9	0.8	0.7	0.8	0.7	0.6	0.7
	0.6	0.6	0.7	0.5	0.7	0.8	0.6	0.7	0.8	0.5	0.6	0.5	0.6	0.7	0.6	0.6	0.8

例2.52　支持向量机不仅可以用于分类问题，也可以用于回归预测。考虑样本，其中$(x_1,y_1),\cdots,(x_l,y_l)$是训练样本，鉴于大多数情况下样本呈非线性关系，估计函数f可按如下方法确定：将每一个样本点用一个非线性函数Φ映射到高维特征空间，再在高维特征空间进行线性回归，从而取得在原空间非线性回归效果，即回归函数f可表示为

$$f(x)=(w\cdot\varphi(x)+b)$$

式中：$x\in R^n$为输入向量；$w\in R^n$为权值矢量；$b\in R$为偏差。

为了得到后两个参数，采用结构化风险最小原则，可以将原来的问题转化为

$$\min\frac{1}{2}\|\omega\|2+C\sum_{i=1}^{l}(\xi_i+\xi_i^*)$$
$$\text{s.t.}\quad y_i-\omega_i-b\leqslant\varepsilon+\xi_i$$
$$\omega_i+b-y_i\leqslant\varepsilon+\xi_i^*$$
$$\xi_i,\xi_i^*\geqslant 0$$

其中：$\|\omega\|^2$是描述函数f的复杂度；$C>0$是常量，用于决定模型复杂度和经验风险的折中度；ξ_i,ξ_i^*是松弛变量，引入的ε是不敏感损失函数。

对于解上述凸优化问题，其核心思想用拉格朗日乘子法把上面的优化问题化为其对偶形式

$$\max\quad -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(x_i\cdot x_j)-\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)$$

$$\text{s.t.}\quad \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0$$
$$\alpha_i,\alpha_i^*\in[0,C]$$

在此对偶式中引入核函数，使得在非线性映射函数φ未知的情况下能够用低维空间的输入数据在高维特征空间完成内积运算。引入的核函数必须满足Mercer定理。

此优化问题可以用二次规则解决。由于支持向量机回归通过最小化避免了数据的欠拟合和过拟合，因此支持向量机是一个更为灵活和通用的解决回归问题的工具。

下面试用支持向量机对春运客流量进行预测。下面为某市火车站2003年春节前后20天的旅客数据。

```
x=[5.3425 10.5679 14.6753 15.3289 14.287 13.6541 12.2313 9.653 11.2345 9.3578
   8.3456 7.8563 6.9682 6.3421 5.8432 6.5437 9.7685 14.3256 15.8645 14.8976]
```

解：

选用前面18个数据作为训练值，后2个数据作为检验。

对于时间序列的预测，一般首先要确定时间窗口m的值，即利用前m个序列值来预测后面的序列值。可以采用求下列自相关系数来确定m。

```
h=mean(x);n=length(x);a1=0;a2=0;
for i=1:n;a2=a2+(x(i)-h)^2;end
for k=1:7;for i=k+1:n;a1=a1+(x(i)-h)*(x(i-k)-h);end;r(k)=a1/a2;end
```

可以从r值确定自相关系数的大小，从而确定m值。在此例中选择m=4，这样就可确定输入数据矩阵：

```
m=4;n=length(x);
for i=m+1:n
for j=1:m;x1(i,j)=x(i-(m-j+1));end
end
x1=x1(m+1:end,:);y=x(m+1:end);
```

然后就可以用支持向量机回归函数进行预测。

为了更好地得到预测结果，首先用遗传算法对支持向量机中的相关参数进行优化。编写下列的适应度函数：

```
function y=mySVM_ga_train(x)
xx=[5.3425 10.5679 14.6753 15.3289 14.287 13.6541 12.2313 9.653 11.2345 9.3578
   8.3456 7.8563 6.9682 6.3421 5.8432 6.5437 9.7685 14.3256];
m=4;[x1,y]=timeinput(xx,m);
ker = struct('type','gauss','width',x(1));C=x(2);nu=x(3); %高斯核函数及相应的参数
svm=mysvmRegression('svr_nu',x1',y,ker,C,nu);              %训练函数
y1=mysvmSim(svm,[5.8432 6.5437 9.7685 14.3256;6.5437 9.7685 14.3256 16.8645]'); %仿真
y=(y1(1)-15.8645)^2+(y1(2)-14.8976)^2;
```

打开遗传算法GUI界面，输入相关的参数就可以计算，一次得到如下的结果：

核参数sigma=8.64，惩罚系数C=19.77，μ=1.52。

利用上述参数进行预测，结果如下，预测精度基本满意。

```
    y=mysvmSim(svm,[5.8432 6.5437 9.7685 14.3256;6.5437 9.7685 14.3256 16.8645]')
    y=16.1269   15.0885
```

如果训练数据能更多些，并利用交叉法进行检验，则预测精度会更好。

第 8 章

关联分析

8.1 概述

目前，关联规则分析已成为数据挖掘领域重要的研究，它主要研究数据中不同领域之间的关系，找出满足给定支持度和可信度阈值的多个域之间的依赖关系。即相关性、关联关系、因果关系。关联规则模式属于描述型模式，发现关联规则的算法属于无监督学习的方法。

R.Agrawal 等于 1993 年提出挖掘顾客交易数据库中项集间的关联规则问题，之后许多关联规则的挖掘问题得到了大量的研究，包括算法、效率等问题。如今，关联规则已得到了广泛的应用，如商品货架设计、附加邮递、目录设计、追加销售、仓储规划以及基于模式对客户进行划分等。

关联规则是发现交易数据库不同商品（项）之间的联系，通过这些规则找出顾客购买行为模式，如购买了某一商品对购买其他商品的影响，发现这些的规则可以应用于商品货架设计、货存安排以及根据购买模式对用户进行分类。现实中，这样的例子很多。最典型的例子是“一些顾客在买婴儿用品时，也同时买啤酒”。一般地，若设商品排列模式为 $x_1,x_2,\cdots,x_n$，第 i 个顾客购买物品为 $x_{i1},x_{i2},\cdots,x_{in}$，同时出现的频率较大，则可以考虑将这两个模式的商品摆放在一起，甚至对这两类商品的模式促销作同一策划。特别是，可以将研究两种不同模式商品同时出现的可能性推广为不同模式商品与分类结果的可能性，推广为多类不同模式的前后多个时间段出现可能性的研究。如电子商务发展的影响因素与发展水平（结果）的相互关系的可能性研究等。

8.1.1 关联规则的主要概念

设 D 是一个事务数据库，其中每一事务 T 由一些项目构成，并且都有一个唯一的标识（TID）。

项目的集合简称为项目集，含有第 k 个项目的项目集称为 k_项目集。

项目集 X 的支持度（support）是指在事务数据库 D 中包含项目集 X 的事务占整个事务的比例，记为 sup（X），看作是项目集 X 在总事务中出现的概率，一般定义为

$$(X) = P(X) \approx \frac{X\text{出现次数}}{\text{事务总数}}$$

支持度是对关联规则重要性（或适用范围）的衡量标准。支持度说明了规则在所有事务中代表性有多大。显然，支持度越大，代表性越大，关联规则也越重要，应用越广泛。由于项目数通常很大，所以，在实际应用中支持度的数据一般都很小。

可信度（Confidence）是指在事务数据库 D 中，同时含项目集 X 和 Y 的事务与含项目集 X 的事务的比，即 $\sup(X \cup Y)/\sup(X)$，看作是项目集 X 的出现，使项目集 Y 也出现这一事务在总事务中出现的频率，一般定义为

$$Conf(Y \mid X) = P(Y \mid X) = \frac{P(YX)}{P(X)} \approx \frac{XY\text{出现次数/ 事务总数}T}{X\text{ 出现次数/事务总数 }T}$$
$$= \sup(X \cup Y)/\sup(X) = \frac{XY\text{出现次数}}{X\text{ 事务总数}}$$

可信度是对关联规则的准确度的衡量。例如对可信度很高但支持度却很低的关联规则来说，它的实际应用价值很小，因而该关联规则的发现也不值得重视。

项目集长度为 k 的子集称为 k 子项目集。如果一个项目集不是任何项目集的长则称此项目集为极大项目集。如果项目集的支持度大于用户指定的最小支持度（min_sup），则称此项目为频繁

项目集（Frequent Item Set）或大项集（large Item Set）。

关联规则可形成化表示为 $X \Rightarrow Y$，它的含义是 $X \cup Y$ 的支持度 $\sup(X \cup Y)$ 大于用户指定的最小支持度 min_sup，且可信度 conf 大于用户指定的最小可信度 min_conf。关联规则挖掘就是在事务数据库 D 中找出满足用户指定的最小支持度 min_sup 和最小可信度 min_conf 的所有关联规则，据此关联分析可分为两个子问题：

（1）找出事务数据库中所有的大项集。

（2）从大项集中产生所有小于大于最小可信度的关联规则。

相对来说，第二个子问题比较容易，目前有关关联规则挖掘的大多数研究主要集中在第一个子问题。关联规则描述虽然简单，但它的计算量很大。假设数据库含 m 个项目，就有 $2m$ 个子集可能是频繁子集，可以证明要找出其一大项集（大频繁集）是一个 NP 问题。

8.1.2　关联规则的种类

关联规则可以按不同的情况进行分类。

（1）基于规则中处理的变量的类型，关联规则可以分为布尔型和数值型。

布尔型关联规则处理的值都是离散的、种类化的，它显示了这些变量之间的关系；而数值型关联规则可以和多维关联或多层关联规则结合起来，对数值型字段进行处理，将其进行动态的分割，或者直接对原始的数据进行处理，当然数值型规则中也可以包含种类变量。

例如：性别 =“女”→职业 =“秘书”，是布尔型关联规则；性别 =“女”→avg（收入）= 2300，涉及的收入是数值类型，所以是一个数值型关联规则。

（2）基于规则中数据的抽象层次，可以分为单层关联规则和多层关联规则。

在单层的关联规则中，所有的变量都没有考虑到现实的数据是具有多个不同的层次的；而在多层的关联规则中，对数据的多层性已经进行了充分的考虑。

例如：IBM 台式机→Sony 打印机，是一个细节数据上的单层关联规则；台式机→打印机，是一个较高层次和细节层次之间的多层关联规则。

（3）基于规则中的数据的维数，关联规则可以分为单维的和多维的。

在单维的关联规则中，只涉及数据的一个维，如用户购买的物品；而在多维的关联规则中，要处理的数据将会涉及多个维。

例如，啤酒→尿布，这条规则只涉及用户购买的物品。性别：“女”→职业 =“秘书”，这条规则就涉及两个数据字段的信息，是两维上的一条关联规则。

8.1.3　关联规则的价值衡量的方法

当通过合适的算法得出了一些结果时间，就需要对这些数据进行衡量，即判断哪些规则对用户来说是有用的。这于这个问题可以从用户主观的层面和系统客观的层面这两个层面进行衡量。

1. 系统客观层面

虽然关联规则的很多算法都使用“支持度—可信度”的框架。但这样的结构有时会产生一些错误的结果。

例如统计了一定数量的学生早晨的运动类型，得到的结果是 55%的学生打篮球，68%的学生晨跑，45%学生晨跑后打篮球。如果设最小支持度为 40%，最小可信度为 60%，可以得到以下的规则：打篮球→晨跑。但这个规则其实是错误的。相反其否定规则：打篮球→（不）晨跑可能更精确。

可以引入“兴趣度”来修剪无趣的规则，即避免生成“错觉”的关联规则。一般一条规则的兴趣度是在基于统计独立性假设下真正的强度与期望的强度之比。然而在许多应用中已发现，只要仍把支持度作为最初的项集产生的主要决定因素，那么要么把支持度设得足够低以使得不丢失任何有意义的规则，或者冒丢失一些重要规则的风险，都可以得到正确的结果。

2. 用户主观层面

虽然规则的产生与算法有关，但一个规则的有用与否最终取决于用户的决定。所以在实际应用中，应将用户的需求和系统更加紧密结合起来。可以采用一种基于约束的挖掘来完成这个目的，它包含以下几个方面。

（1）数据挖掘。用户可以指定对哪些数据进行挖掘，而不一定是全部的数据。

（2）指定挖掘的维和层次。用户可以指定对数据哪些维以及这些维上的哪些层次进行挖掘。

（3）规则约束。可以指定哪些类型的规则是有用的维。通过引入一个模板，当一条规则匹配模板时，可以确定这条规则是令人感兴趣，而哪些则不然。

8.2　Apriori 关联规则算法

Apriori 算法是一种以概率为基础的挖掘布尔型关联规则频繁项集的算法。该算法利用由少到多、从简单到复杂的循序渐进方式，搜索数据库的项目相关关系，并利用概率的表示形成关联规则。它的主要思想是利用“在给定的事务数据库 D 中任意频繁项集的子集都是频繁项集；任意弱项集的超集都是弱项集”这一原理，对事务数据库进行多次扫描，从而找到全部的频繁集。在此过程中，可以利用 Apriori 特性以判断项集是否为频繁集。Apriori 特性是指如果一个拥有 k 个项目的项目集 I 不满足最小支持度，则项目集 I 不是一个频繁集，如果往 I 中加入任意一个新的项目得到一个拥有 $k+1$ 项目的项目集 I' 也必定不是频繁集。

Apriori 算法可大致分为两步：

（1）连接（类矩阵运算）。即通过将两个符合特定条件的 k 项频繁项作连接运算，从而寻找 k+1 项频繁集，而这些频繁集是发现关联规则的基础。

（2）剪枝（去掉不必要的中间结果）。在判断一个项目是否为频繁集时，如是采用对数据库进行扫描计算的方法。当频繁集很大的时候，计算是低效率，而剪枝就是通过引入一些经验性或经数学证明的判定条件，来免除一部分不必要的计算步骤，提高算法效率。

Apriori 算法的主要步骤：

（1）制定最小支持度及最小可信度。

（2）首先扫描数据库产生候选项目集，若候选项目集的支持度大于或等于最小支持度，则该候选项集为频繁项目集。

（3）在运算过程中，首先由数据库读入所有的事务数据，得到候选 1 − 项集集合 C_1 及相应

的支持度数据，通过将每个 1 – 项集的支持度与最小支持度比较，得到频繁 1 – 项集合 L_1，然后将这些频繁 1 – 项集两两连接，产生 2 – 项集合 C_2。

（4）然后再次扫描数据库得到候选 2 – 项集合 C_2 的支持度，将 2 – 项集的支持度与最小支持度比较，确定频繁 2 – 项集。类似地，利用这些频繁 2 – 项集 L_2 产生候选 3 – 项集和确定频繁 3 – 项集，以此类推。

（5）反复扫描数据库，与最小支持度比较，产生更高项的频繁项集合，再结合产生下一级候选集，直到不再结合产生新的候选项集为止。

Apriori 算法的缺陷主要是用时较长，特别是数据库数据较多时。针对这个不足，目前有不少的改进方法。如哈希方法、减少事务数据的方法等。

哈希方法可用于减少生成候选 k – 项集。例如在生成候选 2 – 项集时，不采用对频繁 1 – 项集进行两两连接，而直接对数据库进行扫描。每当扫描一条事务数据时，将事务数据中出现的可能候选 2 – 项集通过哈希函数放入到哈希桶中并修改相应的桶的读数器。在读取完全部的事务数据后，可根据最小支持度检查每个哈希桶的读数器，于是可以直接排除一部分未能达到最小支持度的候选频繁集，因为候选频繁的生成是基于事务数据，因此利用该技术可避免生成支持度为 0 的候选集。然而，算法需要耗时一定的内存空间记录每个哈希桶中的全部候选 2 – 项集内容，在数据库非常庞大时会面临资源不足的风险，此外当一个哈希桶存放的候选 2 – 项集有多种时，对频繁 2 – 项集的判断是相对复杂的，这也是哈希方法的不足之处。

而减少事务数据的方法是假设一个事务数据不能支持任一个 k – 项频繁集，那么它也必不能支持任一个 $k+1$ 项频繁集。该方法在为确定 k – 项频繁集进行数据库扫描的同时，标识每一条数据是否能支持最小一个 k – 项频繁集，在数据库扫描结束后，将不能支持最小一个 k – 项频繁集的事务数据在数据库中进行删除，从而减少了算法下一次扫描数据库所需的时间。

8.3 基于分类搜索的关联规则算法

8.3.1 基于分类搜索的关联规则算法特点

基于分类搜索的关联规则算法具有以下的特点。

1. 分类特点

对于任何项目集 X，若 X 的项目数 length（X）为 k，则 X 属于第 k 类项目集集合，简记为 KI（k）。因此对于事务数据库 D，可以按照事务 T 的项目数归类到 Trade（k）中，并计算出事务 T 出现的次数 count（k），其中 $Trade(k)=\{T \mid length(T)=k, T \subset \text{数据库}D\}$。

2. 搜索特点

由于任何频繁集 F 都是数据库 D 中某个事务集 T 的子集 T_i，即对于任一频繁集 F，至少存在一个 T，使得 $F \subset T$，即 $\sup(F)=\sum_{i=1} count(T_i)$，其中 T_i 为包含 F 的事务集，特别地，若事务 T 为频繁集，则其子集也是频繁集，所以对于项目集 X，若存在事务 T，使 $X \subset T$，当 T 为频繁集，X 也是频繁集。称此为频繁集的充分条件。

3. 存储特点

基于分类的搜索算法将搜索的结果存放在频繁集合或是候选集合中，并通过不断扩充频繁集集合和更新候选集集合，最终得出所有频繁集。由频繁集的封闭性，当频繁集中的项为 1－项子集时其也为 1－项频繁集，称此为频繁集的必要条件。

设 R 为所有 1－项频繁集｛I_{ni}｝的并集，$R=\{I_{n1},I_{n2},\cdots,I_{nm}\}$，称 R 为基础解，由必要条件可知，所有频繁集都是 R 的子集，定义频繁集集合

$$\mathrm{KF}=\{X\mid \sup(X)=\sum_{i=n1} count(T_i)\text{，其中 } X\subset T_i,\mathrm{support}(X)<\min\sup,X\subset R\}$$

定义候选集集合

$$\mathrm{KH}=\{X\mid \sup(X)=\sum_{i=n1} count(T_i)\text{，其中 } X\subset T_i,\mathrm{support}(X)>\min\sup,X\subset R\}$$

基于分类的搜索需要将 KF 按项目数归类为第 k 类频繁集集合和第 k 类候选集集合，分别记为 $\mathrm{KF}(k)=\{X\mid length(X)=k,X\in\mathrm{KF}\}$，$\mathrm{KH}(k)=\{X\mid length(X)=k,X\in\mathrm{KH}\}$。称基于分类搜索的关联规则算法为 KFH 算法。

8.3.2 算法流程与实现

基于分类搜索的关联规则算法过程如下。

（1）对指定数据库，输入最小支持度，计算 1－项频繁集并得出基础解 R 和 KF（1）、KH（1）。

（2）从最多项数 maxk 开始搜索。

（3）在 Trade（k），对于事务 T，若 T 属于 KF 中，则不分解；否则，计算 support（T）为 count（k）与中 T 的支持度之和。若 support（T）>minsup，则将 T 的子集分别放进 KF（i），i=2,3,⋯,k，否则，将 T 放进 KH（k）中，若有子集属于 KF，则放进 KF 中，否则放进 KH 中，更新 KH 中的支持度，KH 中支持度不小于 minsup 的候选集放进 KF 中。循环下去，直到对整个数据库搜索完毕。

（4）对于得到的频繁集，访问数据库，求得其支持度。

8.3.3 数据更新实现

传统算法没有很好地解决 minsup 更新与数据新增的问题。对于新的 minsup 或新的数据，都要重新求解频繁集，导致计算量增大。下面的算法可以很好地解决这个问题。

设集合 $\mathrm{FH}(R)=\mathrm{KF}\cup\mathrm{KH}=\{X\}$ 存在 T，$X\subset T,T\subset R$ 存在，FH（R）为基础解 R 在数据库中存在的所有子集的集合，令

频繁集 $\mathrm{KF}=\{X\mid \mathrm{support}(X)\geqslant\min\sup,X\in\mathrm{FH}\}$

候选集 $\mathrm{KF}=\{X\mid \mathrm{support}(X)<\min\sup,X\in\mathrm{FH}\}$

对于最小支持度的改变或新增数据。KFH 算法的更新过程为

（1）当 minsup 改变时，设基础解变为 R_1，令

$$\mathrm{PH}(R_1\times R_2)=\{X\mid \exists T,X\subset T,X\cap R_2\neq\varnothing,X\subset(R_1\cup R_2)\}$$

式中：R_2 表示原来的基础解；PH（$R_1 \times R_2$）表示基础解为 $R_1 \cup R_2$，即 PH（$R_1 \times R_2$）必含有 R_2 的项，且在数据库中存在的所有子集的集合。若

$$\mathrm{FH}(R_1)=\begin{cases}\mathrm{FH}(R_1)=\{X \mid X \in \mathrm{FH}(X, X \subset R_1)\}, R_1 \subseteq R \\ \mathrm{FH}(R_1)=\{X \mid X \in (\mathrm{FH}(R))+\mathrm{PH}(R \times R_2), X \subset R_1\} \\ R_1=R \cup R_2, R_2 \cap R=\varnothing \\ \mathrm{FH}(R_1)=\{X \mid X \in \mathrm{FH}(R_2))+\mathrm{PH}(\mathrm{PH}(R_2 \times R_3))\} \\ R_1=R_2 \cup R_3, R_2 \subset R=\varnothing\end{cases}$$

则当新的基础解 R_1 包含原基础解 R 时，FH（R_1）也包含 FH（R），所以只要基础解的项不增加，则无须访问数据库就可直接得出新的 KF 和 KH；若出现新的项，则只需要寻找 PH 就可以得出新的 KF 和 KH。

（2）当数据新增时，若新增的项目集集合为 $D'=\{x_1', x_2', \cdots, x_n'\}$，新的基础解为 R_1，则新的 $\mathrm{FH}_{D+D'}=\mathrm{FH}_D(R_1)+\mathrm{FH}_{D'}(R_1)$，由频繁集 KF 和候选集 KF 的计算公式可以求出新的 KF 和 KH，所以新增数据的频繁集更新与改变 minsup 相比只是增加求解 $\mathrm{FH}_{D'}(R_1)$ 的过程。

8.4 时序关联规则算法

序列模式挖掘是指挖掘相对时间或其他模式出现频率高的模式。例如顾客在出租书店租的目录和顺序上表现出来的规律即为一种时序关联规则。对于时序关联规则的挖掘中同样可以采用 Apriori 特性。

给定一个顾客事务（交易）的数据库 D，每一个事务都是由下列字段组成：客户标识（ID）、事务时间（time）及在事务中所购买的商品项目（items）。在同一时间不存在一个顾客多于两个以上的事务发生，在事务中不考虑所购买项目的数量，即只关心一个项目是否被购买。

一个项目集是一个非空的项目的集合，一个序列是由若干个项目集组成的有序的队列。将项目集映射到一个连续的整数集，定义项目集 s_i 为（$i_1 i_2 \cdots i_m$），其中 i_j 是一个项目，则一个序列 $s=<s_1 s_2 \cdots s_n>$是由 s_j 组成的有序的集合。

如果存在整数 $1 \leqslant i_1 < i_2 < \cdots < i_n$，且 $a_1 \subseteq b_{i1}, a_2 \subseteq b_{i2}, \cdots, a_n \subseteq b_{in}$，则称一个序列$<a_1 a_2 \cdots a_n>$属于$<b_1 b_2 \cdots b_m>$，用符号∠表示“被包含于”关系。

在一个序列集合中，若一个序列 s 不被任何其他序列所包含，则称序列 s 是极大的。

一个顾客的所有事务放在一起可看作是一个序列，其中每一个事务对应着一个项目集，而且事务的队列按事务发生的时间升序排列，称这种队列为顾客队列。

序列的长度是序列中的项目集的个数。一个长度为 k 的序列称为 $k-$项序列，如果一个序列 x 的全部项目集是两个序列 y、z 的项目集的并集，称序列 x 是由序列 y 与 z 拼接成的序列，记作 $x=y \cup z$。

如果一个序列 s 被包含于一个顾客的顾客序列中，则该顾客支持序列 s。一个序列$<a_{i1} a_{i2} \cdots a_{in}>$的支持度是支持该序列的顾客数与总顾客数之比，即

$$\sup(<a_{i1} a_{i2} \cdots a_{in}>)=P(a_{i1} a_{i2} \cdots a_{in}) \approx \frac{\sum_{i=1}^{N} C_i}{\text{总顾客数}}$$

其中：C_i是按模式<$a_{i1}a_{i2}\cdots a_{in}$>购买的第 i 个顾客，即 C_i 先购买 a_{i1}，再购买 a_{i2}，以此类推，最后购买 a_{in}。

一个序列<$a_{i1}a_{i2}\cdots a_{in}$>的可信度是指 a_{i1}、a_{i2}、…、a_{in} 模式的规则

$$a_{i1}\Rightarrow a_{i2}\Rightarrow\cdots\Rightarrow a_{in}$$

出现可能性（频率），可以定义为

$$\begin{aligned}Conf(<a_{i1}a_{i2}\cdots a_{in}>) &= P(a_{i1},a_{i2},\cdots,a_{in})\\ &= P(a_{i1})\times P(a_{i2}/a_{i1})\times\cdots\times P(a_{in}/a_{in-1}\cdots a_{i2}a_{i1})\\ &\approx \frac{\text{支持}a_{i1}\text{的顾客}}{\text{总顾客数}}\times\frac{\text{支持}a_{i1}a_{i2}\text{的顾客}}{\text{支持}a_{i1}\text{的顾客}}\times\cdots\times\frac{\text{支持}a_{i1}a_{i2}\cdots a_{in}\text{的顾客}}{\text{支持}a_{i1}a_{i2}\cdots a_{in-1}\text{的顾客}}\\ &\approx \frac{\text{支持}a_{i1}a_{i2}\cdots a_{in}\text{的顾客}}{\text{总顾客数}}\end{aligned}$$

一个满足最小支持度的项目集称为最大项目集，一个满足最小支持度的序列称为大序列。

给定一个顾客事务的数据库 D，序列模式的数据挖掘的问题就是在事务数据库中发现由满足由用户给定的最小支持度的极大的序列，每一个这种极大的序列代表一个序列模式。

时序关联分析有 Apriori－Gen 算法，它可以分为以下几个阶段：

（1）排序。即把以时间作为标识的事务数据库 D 转换为以顾客号作为标识的序列数据库，每一顾客唯一对应一个项目集表示的序列模式。

（2）对序列数据库应用 Apriori 方法求 k 项大项目集（频繁集）。

（3）由 k 项大项目集形成候选序列。

（4）由候选序列应用 Apriori 方法求大序列。

（5）由大序列求最大序列。

8.5 多值属性关联规则算法

由于事务数据中的项目信息是布尔型的，在此基础上发展起来的传统关联规则是针对布尔型数据设计的，因此对于多值关联规则问题需要用多值属性关联规则算法解决。

多值属性可分为数值属性和类别属性。前者如年龄、价格等，可以是连续的，也可以是离散的；后者如品牌、制造商等，只能取有限个属性值。

多值属性的关联规则主要分为以下三类。

8.5.1 静态离散属性关联规则

该方法对数值属性的处理方法是对属性的意义进行研究，结合属性取值的现实意义，预先将属性的值域划分成若干个区间，然后按照这个划分区间对属性值进行离散化，使其从数值型属性转变成类别属性。例如可以将年龄属性按不同的年龄段分成童年、青年、中年和老年四种年龄阶段。

在进行属性的转换过程中，要结合数据集的属性值的分布特点，否则会导致无法挖掘出有意义的关联规则。转换过程中需要注意如下问题。

（1）标称属性不能取值过多，否则因没有足够支持度支持，会无法发现任何关联规则。

（2）连续属性离散区间划分要适当。区间划分太窄会导致不满足支持度，而无法发现关联

规则。

8.5.2　动态离散关联规则

该方法的区间划分并不通过事先的定义，而是取决于数据的分布情况。划分的区间数也是不确定的，并且在挖掘的过程中可能会根据需要而将一些相邻的区间进行合并以获得更强的关联规则。

某些量化属性的值域范围是相对固定的，例如如果要将年龄离散化，只需将值域放大到[0,150]，则区间划分可以在这个确定的区域中进行。但是某些量化属性如收入，其值域可以非常宽广，而且在不同的数据库，其数据的分布可以是不均匀的。对于这种量化属性可以根据其分布特点进行动态划分，在值域宽泛的时候区间宽度会随着值域扩大，而在值域比较狭小时区间的划分会更加细致，而不会丢失属性信息。这个划分过程称为分箱。常用的分箱策略有以下三种。

（1）等宽分箱　每个箱的区间长度相同。

（2）等深分箱　每个箱赋予大致相同个数的元组。

（3）基于同质的分箱　箱的确定取决于使得每个箱的元组分布趋于一致。

8.5.3　基于距离的关联规则

动态离散方法能够根据属性数据的分布进行离散化，但是这种划分可能不完全符合区间数据的语意。一种可行的划分方法是聚类方法，通过将该属性的全部数据进行聚类，从而得到若干个类别，再根据类中数据的极大、极小值来确定区间边界。

在将属性值采用不同方法离散化后，这些方法都可以共享一个相同的求解框架，即可将多值关联规则问题转化为布尔型关联规则问题，然后再利用已有的挖掘布尔型关联规则的方法得到有价值的规则。若属性为类别属性，则先将属性值映射为连续的整数，并使意义相近的取值相邻编号。

算法求解过程如下。

（1）将划分后的属性区段[*lk*，*rl*]或属性值映射成序对<*A*，*k*>，进而映射为布尔属性 *A*（*m*），所有这样的属性构成项集。

（2）从项集中寻找所有有价值的项，构成频繁集。有价值的项是指支持它的交易的数量超过给定的最小支持度。

（3）在频繁集中迭代地搜索出组合后的支持度超过给定阈值的两个项，将其组合并加入频繁集中，如果是相同属性的相邻区段，则进一步合并。

（4）应用频繁集产生关联规则，如果 *ABCD* 和 *AB* 都是频繁集，则判定规则 $AB \Rightarrow CD$ 是否成立，是通过计算可信度 $Conf = \dfrac{\mathrm{supp}(ABCD)}{\mathrm{supp}(AB)}$ 是否超过最小可信度来决定的。如果成立，则规则成立。

（5）确定有价值的关联规则作为输出。

8.6　增量关联规则算法

在有关关联规则的应用中，为了找到真正有价值的规则，需要不断调整两个基本的变量即最

小支持度和最小可信度。

给定交易数据库，一个项目集的支持度可以认为是所有包含该项目集的交易的数目，设最小支持度为 s，L_k 为所有频繁 k－项集的集合，$k=1,2,\cdots,m(1)$。这里 $m(1)$为所有频繁项目集长度中的最大者。同样对于新的支持度 s'，设 $L(k)$为所有频繁 k－项目集的集合，$k=1,2,\cdots,m(2)$。对于每一个项目集都有一个域 count 用来保护它的支持度计数。

当最小支持度发生改变时，可以分为如下两种情况。

（1）$s'>s$，原有的一些频繁项目集可能最小支持度。

（2）$s'<s$，原有的一些非频繁项目集可以获得最小支持度。

假设原支持度 S_1，新支持度 S_2，首先通过扫描得到 S_2 下的新频繁集 L_2（1），且 L_2（1）与 L（1）不相交，L_1（1）为 L_2（1）与 L（1）的并集，之后就得到三类频繁 k－项目集 C_1（k），C_2（k），C_3（k），对这 C_1（k）和 C_2（k）进行 Apriori 算法中 apriori－Gen 函数生成，对 C_3（k）通过剪枝可以求得。

8.7 基于关联规则的分类算法

分类是数据挖掘领域中最重要的应用之一。对于不确定性问题的关联分类规则需要关联规则分类算法。

基于关联规则的分类问题定义如下：

假设数据集 D 有 m 个属性分别为 $A_1,A_2,\cdots,A_m$，数据集的规模为｜D｜，C 表示数据的类别标志。相应地，属性 A_i 和类别 C 的取值可表示为 a_i 和 c，则一条数据可表示为（$a_1,a_2,\cdots,a_m$,c）项目集 Itemset 定义为若干个属性的取值的集合$<a_{i1},a_{i2},\cdots,a_{ir}>$，$q\leqslant m$。一条分类规则 r 可以表示为一个项目集对应于一个类别值，即$<a_{i1},a_{i2},\cdots,a_{ir}>$。Occr（$r$）表示数据集 D 中与构成规则 r 的项目集$<a_{i1},a_{i2},\cdots,a_{ir}>$相匹配的数据量。Supp（$r$）表示规则 r 的支持度，即数据集 D 中与构成规则 r 的项目集相匹配且类别标志与规则 r 的类别一致的数据量。Conf(r) = supp(r)/Occr(r)表示规则 r 的置信度。minsup 和 minconf 分别表示关联规则分类中的最小支持和最小置信度。规则集 R 表示分类规则 r 的集合，其中 $\forall r\in R$ 满足 supp(r)>minssup 且 conf(r)>minconf。对于规则 $r<a_{i1},a_{i2},\cdots,a_{iq},c\subset R$，可以看作 $a_{i1},a_{i2},\cdots,a_{iq}\Rightarrow c$，假定某一数据记录的属性取值同时满足 $a_{i1},a_{i2},\cdots,a_{iq}$时，则它属于 c 类的置信度为 conf(r)。关联规则分类算法是在基于样本数据集 D，寻找分类规则集 R，并按照某种策略将其中的分类规则有序地组织起来，从而建立分类器模型。在给定一组未知类别的数据时，通过在分类器中的规则集中寻找置信度高并且与数据最优匹配的规则，将匹配规则指向的类别作为未知数据的类别。

根据上述的定义可以发现，关联分类算法实际上是对多值属性关联规则问题的进一步挖掘。在多值属性关联规则问题上需要关注的是各自属性值之间的关系，各个属性的地位是否平等，而对于关联分类问题，把类别标志也作为一个属性来看待，从而通过关联规则方法挖掘属性之间的潜在关系。关联分类算法在寻找到关联规则的基础上，挑选出那些与类别属性密切相关的规则，并按某种逻辑顺序把这些规则整合成为一个有机整体，即分类器。通过将未知样本的属性和分类规则进行匹配，将最匹配的规则的类别指派给未知样本。从而实现对样本进行分类。

关联规则分类算法通常分为三个相继的环节，即规则生成、规则梳理和分类。规则生成主要沿用关联规则挖掘技术挖掘蕴含分类规则的频繁集，规则梳理则在过滤部分无用规则的基础上采

用某种标准将分类规则组织起来形成分类器模型，分类则利用建立的分类器对未知类别数据进行判别。

现有的关联分类算法包括 CBA、CMAR、MCAR、GARC 等，其中 CBA 算法是最早用于关联分类的一个算法，它在规则生成环节基本上平移了基于 Apriori 的多属性关联规则算法来寻找分类规则，规则梳理环节在获得分类规则集的基础上，按照规则的置信度、支持度、规则长度等标准，将所有的规则按照线性的顺序组织起来，形成一个分类器，而在分类环节中，按照分类器中的规则的先后次序将规则的条件与未知类别进行匹配，从而找到最优的规则，完成对样本的分类。

8.8　模糊关联分类算法

8.8.1　属性的模糊划分

（1）数值型属性的模糊划分

三角函数常被用于定义数值型属性的模糊划分。当属性 a 取值范围划分为 K 类时，属性 a 隶属于 K 类中的第 i_1 类的隶属度函数为

$$\mu_{K,i_1}^{a}(x)=\max\{1-|x-a_{i_1}^{K}|/b^{K},0\}$$

式中：$a_{i_1}^{K}$ 是 i_1 类中心，$a_{i_1}^{K}=m_i+(m_a-m_i)(i_1-1)/(K-1)$，$b^{K}=(m_a-m_i)/(K-1)$，$m_a$ 是属性 a 取值范围里的最大值，而 m_i 是属性取值范围的最小值，b^K 是对应类边界。

对于划分中心的选择，可以先结合建模样本对每个属性按模糊区间数进行聚类，找到相应的类中心并作为属性模糊区间的中心，相应地，可取两个类别的最靠近中心的点的距离中点为边界。

（2）离散属性的模糊划分

如果一个离散型属性的值共有 n 种，那么这种属性可以被界定为 n 种模糊划分，将每一个属性可以对应于一个整数，则对于属性值 $A_{n,i_m}^{x_m},1\leqslant i_m\leqslant n$ 可以被定义为第 i_m 个划分（$i_m-\varepsilon$，$i_m+\varepsilon$）而隶属度函数为 $\mu_{n,i_{m1}}^{a}(x)=1,i_m-\varepsilon\leqslant x\leqslant i_m+\varepsilon,\varepsilon\to 0$ 。

8.8.2　模糊关联的定义

根据属性值的模糊划分，可以给出模糊意义下的支持度和置信度的定义。

给定数据集 D，｜D｜＝n，对属性 a，隶属度函数为 $\mu_{K,i_j}^{a}(X_p)=\max\{1-|x_{a,p}-a_{i_j}^{K}|/b^{K},0\}$，其中 $a_{i_j}^{K}=m_i+(m_a-m_i)(i_j-1)/(K-1)$，$b^{K}=(m_a-m_i)/(K-1)$，$a_{x,p}$ 是样本 $X_p\subset D$ 在属性 a 上的值，则属性值 A_{k,i_j} 的模糊支持度为

$$\mathrm{supp}_{fuzzy}=\sum_{X_p}^{D}\mu_{K,i_1}^{a}(X_{a,p})/|D|$$

模糊置信度为

$$\mathrm{conf}_{fuzzy}=\frac{\sum_{X_p}^{D}\mu_{K,i_1}^{a_1}(X_{a_1,p})\bullet\mu_{K,i_1}^{a_2}(X_{a_2,p})}{\sum_{X_p}^{D}\mu_{K,i_1}^{a_1}(X_{a_1,p})\bullet\sum_{X_p}^{D}\mu_{K,i_1}^{a_2}(X_{a_2,p})}$$

模糊关联分类是在数据集 D 中依据模糊支持度和模糊置信度的计算公式寻找模糊支持度和置信度大于相应阈值 minsupp 和 minconf 的模糊分类规则，并构造模糊关联分类器，对未知样本以置信度进行模糊分类。

8.9 关联规则的评价

在实际应用中，由于数据库的数据量和维数都很大，很容易产生数以百计的关联规则，如何从中筛选出最有价值的规则显得非常重要。为此需要建立一组广为接受的评价关联规则质量的标准。常用的评价标准主要有两种：一是基于统计学的客观度量（如基于支持度—置信度框架）；另一种是通过主观论据建立的主观度量。

8.9.1 支持度—置信度框架

现有关联规则算法大部分都使用支持度—置信度框架来除去没有意义的规则。支持度的度量反映了关联规则是否具有普遍性，支持度高说明这条规则可能适用于数据中的大部分事务。置信度的度量则反映了关联规则的可靠性，置信度高说明如果满足了关联规则的前件，同时满足后件的可能性也非常大。尽管最小支持度和置信度阈值有助于排除大量无意义规则，但是仍然会产生一些没有价值的规则。支持度的缺点在于许多潜在的有意义的规则由于包含支持度小的项而被删除；而置信度忽略了规则后件中项集的支持度，可能出现误导的强关联规则。因此，由支持度—置信度度量得出的强关联规则不一定是有意义的规则。

例如表 8.1 所示为早餐麦片的销售商调查在校 5000 名学生早晨进行的活动，假定支持度为 40%，置信度为 60%。则关联规则｛打篮球｝→｛吃麦片｝的支持度为 2000/5000=40%，置信度为 2000/3000=67%。这条规则是强关联规则，表明打篮球的学生通常也会吃麦片。但是所有学生中吃麦片比例为 75%，要大于 67%。这说明一个学生如果打篮球，那么他吃麦片的可能性就从 75%下降到了 67%，而且｛不打篮球｝→｛吃麦片｝的可能性为 1750/2000 = 87.5%。因此，尽管规则｛打篮球｝→｛吃麦片｝有着较高的置信度，却是一个误导，因为打篮球反而会抑制早餐吃麦片。

表 8.1 早餐与运动的调查结果

	打 篮 球	不打篮球	
吃麦片	2000	1750	3750
不吃麦片	1000	250	1250
	3000	2000	5000

为了降低支持度和置信度度量的局限性，可以在它们的基础上增加相关性的度量。相关性度量可采用提升度（亦称兴趣度）、相关系数、余弦度量等方法。

提升度（lift）是一种简单的相关度量。对于项集 A 和项集 B，如果 P(A∪B)=P(A)P(B)，则 A 和 B 是相互独立的，否则存在某种依赖关系。关联规则的前件项集 A 和后件项集 B 之间的依赖关系通过提升度计算：

$$\mathrm{lift}(\mathrm{A},\mathrm{B})=\frac{\mathrm{P}(\mathrm{A}\cup\mathrm{B})}{\mathrm{P}(\mathrm{A})P(\mathrm{B})}=\frac{\mathrm{confidence}(\mathrm{A}\to\mathrm{B})}{\mathrm{support}(\mathrm{B})}$$

提升度可以评估项集 A 的出现是否能够促进项集 B 的出现。如果其值大于 1，表示二者之间存在正相关；小于 1，则二者存在负相关；等于 1，二者之间没有相关性。

根据表 8.1 可以计算出关联规则｛打篮球｝→｛吃麦片｝的提升度为

$$\text{lift}(\{打篮球\} \to \{吃麦片\}) = \frac{\text{P}(\{打篮球\} \quad \{吃麦片\})}{\text{P}(\{打篮球\}) \quad (\{吃麦片\})} = \frac{0.4}{0.6 \times 0.75} = 0.89$$

其值小于 1，说明前后件存在负相关关系，即推广“打篮球”不但不会提升“吃麦片”的人数，反而会减少。

项集间的相关性也可以用相关系数来度量。对于二元变量，相关系数 ϕ 定义为

$$\phi = \frac{f_{11} f_{00} - f_{01} f_{10}}{\sqrt{f_{1+} f_{+1} f_{0+} f_{+0}}}$$

式中：每个 f_{ij} 都表示一个频度计数，f_{11} 表示 A 和 B 同时出现在一个事务中的次数，f_{01} 表示包含 B 但不包含 A 的事务数，f_{1+}表示 A 的支持度计数，f_{+1} 表示 B 的支持度计数。

同样，可计算出“打篮球”与“吃麦片”间的相关系数，其值小于 0，说明存在负相关。

$$\phi = \frac{2000 \times 250 - 1000 \times 1750}{\sqrt{3750 \times 3000 \times 1250 \times 2000}} = -0.23$$

相关性度量还可以用余弦度量，即

$$\cos ine(A,B) = \frac{\text{P(A} \cup \text{B)}}{\sqrt{\text{P(A)} \times \text{P(B)}}} = \frac{\text{support(AÈB)}}{\sqrt{\text{support(A)} \times \text{support(B)}}}$$

8.9.2　基于主观因素的主观度量

主观度量的评估标准主要体现在用户和领域知识融合等主观因素，是一项较为困难的任务，需要来自领域专家的大量知识信息。

下面是几种将主观因素加入到关联规则发现的方法。

（1）可视化：这种方法需要友好的环境界面，保持用户参与，允许领域专家解释和检验所发现的规则，并能与数据挖掘系统交互。

（2）基于模板的方法：这种方法允许用户限制挖掘算法的模式类型，只把满足用户指定模板的规则提供给用户，而不是提取所有规则。

（3）主观兴趣度度量：主观兴趣度可以基于领域信息来定义，如概念分层或商品利润等，然后使用这些度量来过滤没有意义的规则。例如规则{黄油}→{面包}可能不是十分有趣的，尽管有很高的支持度和置信度，但是它表示的关系显而易见。另一方面，规则{Diaper}→{Beer}是有趣的，因为这种关系十分出乎意料，并且可能为零售商提供新的交叉销售机会。

8.10　辛普森悖论

在实际的关联分析中，有时在对数据集按照某个变量进行分组后，则之前对整个数据集分析得到的关联规则可能并不适用于分组数据，这种现象就是辛普森悖论。

表 8.2 为某学校的招生数据，可计算出规则{性别＝男}→{录取＝是}的置信度是 209/304=68.8%，而规则{性别＝女}→{录取＝是}的置信度是 143/253=56.5%，说明男生比女生更有可能被录取。

表 8.2　招生录取表

性　别	录　取		总　数
	是	否	
男	209	95	304
女	143	110	253
总数	352	205	557

将招生数据按学院进行分组后，招生录取情况如表 8.3 所示。

表 8.3　分组后招生录取表

学　院	性　别	录　取		总　数
		是	否	
法学院	男	8	45	53
	女	51	101	152
商学院	男	201	50	251
	女	92	9	101

对于法学院：

confidence({性别＝男}→{录取＝是})=8/53=15.1%

confidence({性别＝女}→{录取＝是})=51/152=33.6%

对于商学院：

confidence({性别＝男}→{录取＝是})=201/251=80.1%

confidence({性别＝女}→{录取＝是})=92/101=91.1%

计算表明，对于两个学院，女生更有可能被录取。这与先前由包含两个学院的数据得到的结论刚好相反。即使采用其他度量（如相关性、概率或兴趣因子）也会发现，在所有数据情况下男性和录取之间存在正相关，但在组分数据情况下却存在负相关的情况。得到的这两种截然不同的结论就是辛普森悖论。

辛普森悖论的存在使得在进行相关分析时，有时需要对数据进行适当的分组，才能避免因辛普森悖论产生虚假的模式。例如大型超市的购物篮数据应该依据商品的位置分组，而不同病人的医疗记录应当按照不同的因素（如年龄和性别）分组。

8.11　基于 MATLAB 的关联规则分析

例 2.53　表 8.4 给定一个具有 9 条数据的事务库。假设最小支持度为 0.2，最小置信度为 50%，求大项目。

表 8.4　事务数据集

标　识	项目清单	标　识	项目清单
T001	I1,I2,I5	T006	I2,I3
T002	I2,I4	T007	I1,I3
T003	I2,I3	T008	I1,I2,I3,I5
T004	I1,I2,I4	T009	I1,I2,I3
T005	I1,I3		

解：

根据 Apriori 算法的原理，编程分析本问题。在程序中为了减少对数据库的扫描次数，首先对数据库扫描，得到一个矩阵，其中行表示项，列表示这个项出现在标识中的标识号。如对于本例其矩阵为

```
a=[ 1      NaN    NaN    4      5      NaN    7      8      9
    1      2      3      4      NaN    6      NaN    8      9
    NaN    NaN    3      NaN    5      6      7      8      9
    NaN    2      NaN    4      NaN    NaN    NaN    NaN    NaN
    1      NaN    NaN    NaN    NaN    NaN    NaN    8      NaN];
```

得到这个矩阵后，其后在计算支持度时就无须再扫描数据库，而只需对矩阵进行减法操作即可。

同时，为了使程序能适合不同的情况，本程序可以用三种形式输入，即：一是用数字表示事项；二是用字母表示事项；三是数字序号不是从1开始。如本例可以用下面的其中一种形式输入：

```
x={[1 2 5];[2 4];[2 3];[1 2 4];[1 3];[2 3];[1 3];[1 2 3 5];[1 2 3]};
```

或者：

```
x={[3 4 7];[4 6];[4 5];[3 4 6];[3 5];[4 5];[3 5];[3 4 5 7];[3 4 5]};
```

或者：

```
x={{'I1','I2','I5'}{'I2','I4'};{'I2','I3'};{'I1','I2','I4'};{'I1','I3'};
{'I2','I3'};{'I1','I3'};{'I1','I2','I3','I5'};{'I1','I2','I3'}};
```

对于本例，计算可得到以下结果：

```
>> sup_min=0.2;conf_min=0.5;
>>x={[1 2 5];[2 4];[2 3];[1 2 4];[1 3];[2 3];[1 3];[1 2 3 5];[1 2 3]};
>> y=Apriori(x,sup_min,conf_min)
y=  '5→1  2  conf=1'          %关联规则及相应的置信度
    '1  5→2  conf=1'
    '2  5→1  conf=1'
    '1  2→3  conf=0.5'
    '1  2→5  conf=0.5'
```

```
    '1  3→2  conf=0.5'
    '2  3→1  conf=0.5'
```

例 2.54　当数据库较大时，基本的 Apriori 算法耗时较大，需要进行改进。其中的一个改进方法是将一个大的事务数据库划分为若干个规模较小的事务数据库，并在各个小事务数据库中挖掘出极大频繁集。然后将全部的局部极频繁集汇总起来形成候选全局极大频繁集，最后再一次扫描大数据集计算每个候选全局极大频繁集的支持度，最后可得到全局极大频繁集。

根据例2.52中的数据表，构造一个事务数据库，并根据这个算法的原理挖掘极大频繁集。

解：

根据算法原理，可编制相应的程序。在程序中，支持度的计算是基于一个二值矩阵，其结构与例2.52中的数据矩阵类似，只不过用“1”代表出现，“0”代表不出现。在计算支持度时对矩阵相应的行进行逻辑“或”运算即可。

在划分数据库时，各个小数据库中的数目可以不相等，另外在具体应用时，要根据具体情况（整个数据库大小、内存等）确定小数据库的数目，本例中划分为3个。

```
>>sup_min=0.2;conf_min=0.5;sample=cell(31,1); type=2;
>>x={{'I1','I2','I5'};{'I2','I4'};{'I2','I3'};{'I1','I2','I4'};{'I1','I3'};
{'I2','I3'};{'I1','I3'};{'I1','I2','I3','I5'};{'I1','I2','I3'}};
>>sample=repmat(x,[],3);sample{28}=x{8};sample{29}=x{9};
>>sample{30}={'I1','I2','I6'};sample{31}={'I1','I5','I6'};
>>rule=apriori_divi(sample,sup_min,conf_min,type);
>> rule{1}='I2'   'I5'   '→'   'I1'   '  conf='   '1'
   rule{2}='I1'   'I5'   '→'   'I2'   '  conf='   '0.875'
   rule{3}='I5'   '→'   'I1'   'I2'   '  conf='   '0.875'
   rule{4}='I1'   'I3'   '→'   'I2'   '  conf='   '0.57143'
   rule{5}='I2'   'I3'   '→'   'I1'   '  conf='   '0.57143'
   rule{6}='I1'   'I2'   '→'   'I3'   '  conf='   '0.53333'
```

第 9 章

其他数据挖掘方法

9.1 近邻法

近邻法是在数据挖掘中使用最早的技术之一。其基本思想是为了预测一个记录中的预测值，或在历史数据库中寻找有相似预测值的记录，可以使用未分类记录中最接近的记录值作为预测值，也即相互之间接近的对象会有相似的预测值。

假设有 M 个 $\omega_1,\omega_2,\cdots,\omega_M$ 类别，每类有标明类别的样本 N_i 个（i=1,2,$\cdots$,M），可以规定 ω_i 类的判别函数为

$$d_i(X)=\min_k \| X-X_i^k \|$$

其中：X_i^k 的角标 i 表示 ω_i 类，k 表示 ω_i 类 N_i 个样本中的第 k 个。分类器规则可以写为

$$d_j(x)=\min_i d_i(x), i=1,2,\cdots,M \Rightarrow X\in\omega_j$$

这一决策过程称为最近邻法，也即对未知样本，只要比较与 $N=\sum_{i=1}^{M}N_i$ 个已知类别的样本间的欧氏距离，并将其归类与离它最近的样本类别。

上述方法只根据与未知样本最近的一个样本的类别而决定未知样本的类别，通常称为 1NN 方法。为了克服单个样本类别的偶然性以增加分类的可靠性，可以采用 K-近邻法（K-nearest neighbors，KNN），即考察与未知样本 x 最近邻 k 个样本，这 k 个最近邻中哪一类的样本最多，就将 x 判属哪一类。为了避免近邻数相等，一般 k 采用奇数。另外最近邻样本对于“选票”所起的作用，可以用相应的距离将之赋权

$$V_{总}=\sum_{i=1}^{k}\frac{V_i}{D_i} \quad 或 \quad V_{总}=\sum_{i=1}^{k}\frac{V_i}{D_i^2}$$

式中：V_i 为对于两类问题，当其邻属于第一类时，为“+1”，属于第二类时为“−1”，D_i 为未知样本与第 i 个近邻的距离，k 为最近邻数。当“选票”$V_{总}>0$ 时，则未知样本归入类 1，否则未知样本归入类 2。

为了测试 k 个最近邻样本的风险值，可用下式计算

$$R_i^{(k)}=[1+\frac{1}{k}+a\delta^2(k)]R^*$$

KNN 法无须要求对不同类的代表点线性可分，只要用每个未知点的近邻类来判别就可以；也不需要作训练过程。但它的缺点是没有对训练点作信息压缩，因此每判别一个新的未知点都需要把它和所有已知代表点的距离全部算一遍，因此计算工作量大，对已知代表点太多的情况不甚合适。但正是因为没有作信息压缩，而用全体已知点的原始信息做判据，故有时可得到极好的预报准确率，其效果一般优于或等于其他模式识别方法。

KNN 法中对所有的类选取相同的 K 值，且其选择有一定的经验性。如果能根据每类中样本的数目和分散程度选择 K 值，并当各类的 K_i 选定后，用一定的算法对类中样本的概率进行估计，并且根据概率大小对它们进行分类，将会影响 K 值选择的经验性。ALKNN（Alternative KNN）正是基于这样的思想。

在 AKNN 方法中，以 x_i 与类 g_i 的 K_i 个近邻中最远一个样本的距离 r 为半径，以 x_i 为中心，计算相应的超球的体积，并且认为超球体积越小，类 g_i 在 x_i 处的概率密度越大。其概率密度可用

下式计算

$$P(x_i / g_i) = \frac{K_i - 1}{n[V(x_i / g_i)]}$$

其中：V（x_i/g_i）为类 g_i 的超球体积，该超球中心为 x_i，半径为 r。为了选择 K_i 和相应 r 的计算，可采用欧氏距离，m 维超球体积的一般表达式为

$$V(x_i / g_i) = (2\pi)^{m/2} r^m [m\Gamma(m/2)]$$

其中：Γ 为 gamma 函数。

在实际计算中，上述方程根据 m 的奇偶性可以写成下列两种形式：

当 m 为偶数时：

$$V(x_i / g_i) = (2\pi)^{m/2} r^m [m(m-2)(m-4)\cdots]$$

当 m 为奇数时：

$$V(x_i / g_i) = 2(2\pi)^{(m-1)/2} r^m [m(m-2)(m-4)\cdots]$$

计算时必须对 K_i 进行优化，这样才能对各类概率密度的测试相一致。K_i 值的优化公式可采用下列公式

$$\max \quad g(k_i) = \sum_{t=1}^{n} \ln p(x_{it} / g_i)$$

对样本的分类采用后验概率，其计算公式为

$$P(g_i | x) = P(x | g_i) / \sum_{i=1}^{G} [P(x | g_i)]$$

即样本划归具有最大后验概率的类中。

9.2　K-means 聚类

K-means 聚类是一种实际应用较多的聚类方法，它的核心思想是通过迭代把数据对象划分到不同的簇中，以求目标函数最小化，从而使生成的簇尽可能地紧凑和独立。给定样本集和整数 K，K-means 算法将样本集分割成 K 个簇，每个聚类中心是簇中样本的均值；然后将其余对象根据其与各个簇的中心的距离分配到最近的簇，再求新形成的簇的中心。这个迭代重定位过程不断重复，使得每个簇中所有样本与其中心的距离总和最小，直至目标函数最小化为止。此算法的结果受到聚类中心的个数以及初始聚类中心的选择影响，也受到样本几何性质及排列次序影响。如果样本的几何特性表明它们能形成几个相距较远的小块孤立区域，则算法都能收敛。

算法原理如图 9.1 所示，具体描述如下：

（1）确定分类数目（K）和最大迭代次数。

（2）初始化。随机取 K 个样本作为聚类中心，其余样本中心号为 −1，样本到本类中心的距离为无穷大。

（3）计算其余样本到 K 类中心的距离，并将它归为距离最近的类，到所有样本都归类完毕。计算各个类中心所有样品特征值的平均值作为该聚类中心的特征值。

（4）对每一类中的各个样本，计算它到其他类中心的距离，如果它到某一个类中心的距离

小于它到自身类中心的距离，需要对该样本重新分类，将它归属到距离中心近的类，循环重复所有的样本，直至不再有样本类号发生变化。

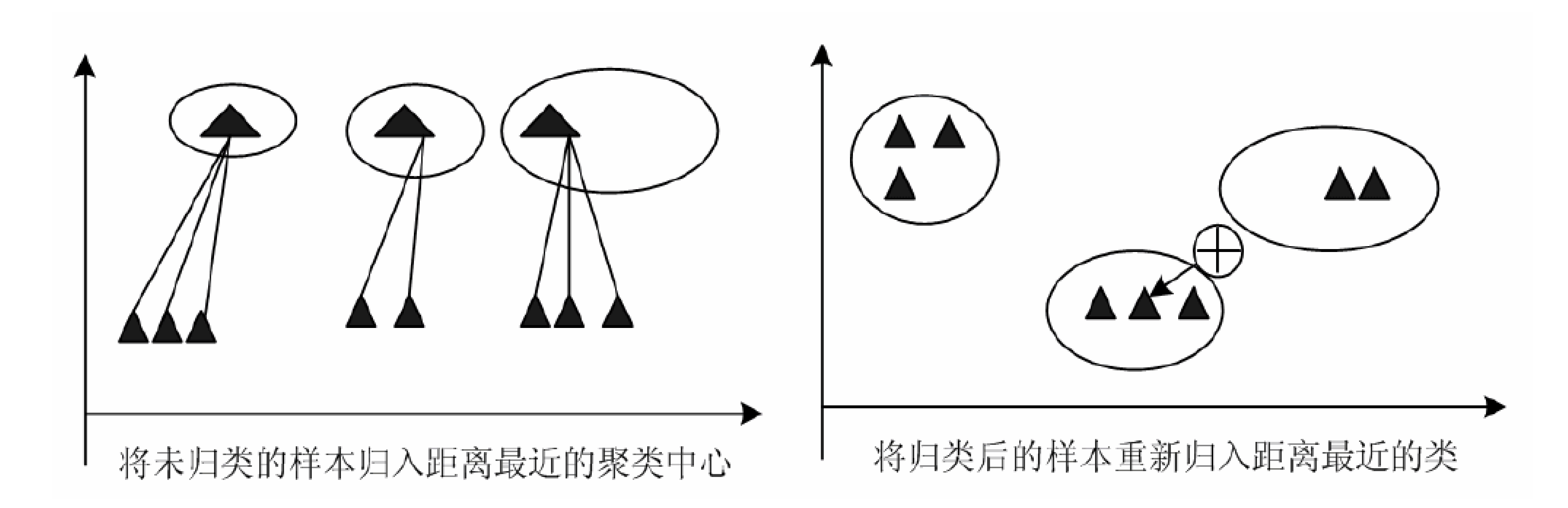


图 9.1 K-means 算法示意图

面对大规模数据集，该算法是相对可扩展的，并且具有较高的效率。算法复杂度为 $O(nkt)$，其中 n 为数据集中对象的数目，k 为期望得到的簇的数目，t 为迭代的次数，算法通常终止于局部最优解。

K-means 法的缺点在于要事先给出期望生成簇的数目，这在某些应用中是不实际的，另外它不适合于发现非凸面形状的簇和大小差异较大的簇，并且该算法对“噪声”和孤立点数据敏感。

可以通过考察簇的分离情况和簇的紧凑情况即轮廓系数来评估聚类效果。其计算步骤如下。

（1）从数据点随机取出第 i 个数据点，并计算该点到 S 簇中其他所有数据点的平均距离 d_i。

（2）计算该点到所有其他簇中所有数据点的平均距离，并找到最小平均距离 Md_i。

（3）计算轮廓系数 $SC_i=(Md_i-d_i)/\text{Max}(d_i, Md_i)$，此值越接近 1，说明该点的划分越好。

（4）将簇中所有点的轮廓系数取平均值，综合所有簇的平均轮廓系数，此值越高的分类方案越优。

在 MATLAB 中利用 silhouette 函数就根据聚类结果绘制轮廓图。

在初始的 k 个均值选择，对象相异度计算、簇均值的计算等方面采取不同的将得到均值算法的很多变形。例如 k-模方法用模代替簇的均值，用新的差异度度量方法处理对象，用基于频率的方法修改簇的模。而 k-原型方法将 k-均值和 k-模算法集成在一起，用于处理含有数值和分类值属性的数据聚类。

K-means 算法采用簇的质心来代表一个簇，质心是簇中其他对象的参照点。因此该算法对孤立点是敏感的，如果孤立点具有极大值，就可能大幅度地扭曲数据的分布。此时可用 K-中心点算法代替 K-means 算法，它选择簇中位置最接近簇中心的对象（即中心点）作为簇的代表点，目标函数仍然可以采用平方误差准则。

K-中心算法的具体描述如下：

（1）确定分类数目（K）和最大迭代次数；

（2）选择 K 个对象作为初始的簇中心；

（3）对每个对象，计算离其最近的簇中心点，并将对象分配到该中心点代表的簇；

（4）随机选取非中心点 O；

（5）计算用 O 代表 O_j，形成新集合的总代价 S（其度量为对象与代表点之间的平均相异度）；

（6）如果 $S<0$，用 O 代替 O_j，形成新的 K 个中心点的集合；

（7）重复（3）~（6），直至不再发生变化。

K-means 算法中距离的计算基于数值型数据，没有明确说明对于分类型数据如何处理。此外，它对于噪声和离群点数据较为敏感。为了克服这些缺点，可以作如下方面的改进。

在初始的 k 个 means 选择，对象相异度计算、簇均值的计算等方面采取不同的将得到 K-means 算法的改进算法。以下即为常用的三种改进措施。

（1）将分类型数据转化为数值型数据，再利用 K-means 算法进行聚类。

对于具有 k 个类别的标称型变量，采用 k 个取值为 0 或 1 的数值型变量共同来表示。例如变量具有三个类别 A、B 和 C，则可以用 100 表示 A 类别，010 表示 B 类别，001 表示 C 类别。

（2）采用适用于纯分类属性数据集的 K-modes 算法和适用于混合属性数据集的 K-prototypes 算法。

K-modes 算法采用众数（取值频率最大的属性值）来表示分类属性，在聚类过程中采用简单匹配来度量分类属性的不相似性。

K-prototypes 算法则是 K-modes 算法和 K-means 算法的结合。

（3）采用适用于混合属性数据集的 K-Summary 算法。

对于聚类分析而言，簇的表示和数据对象之间相似度的定义是最基础的问题，直接影响到数据聚类的效果。

假设数据集 D 有 m 个属性，其中有 m_C 个分类属性和 m_N 个数值属性，用 D_i 表示第 i 个属性取值的集合。

给定簇 C、C_1 和 C_2，对象 $p=[p_1,p_2,\cdots p_m]$ 与 $q=[q_1,q_2,\cdots,q_m]$，$x>0$

① 对象 p,q 在属性 D_i 上的差异度（或距离）$\mathrm{dif}(p_i,q_i)$ 定义为

对于分类属性或二值属性，$\mathrm{dif}(p_i,q_i)=\begin{cases}1\\0\end{cases}$ 或 $\begin{cases}0 & p_i\neq q_i\\1 & p_i=q_i\end{cases}$

对于连续型属性或顺序属性，$\mathrm{dif}(p_i,q_i)=|p_i-q_i|$

② 两个对象 p,q 间的差异程度（或距离）$d(p,q)$ 的定义为

$$\mathrm{dif}(p_i,q_i)=\left(\sum_{i=1}^{m}\mathrm{dif}(p_i,q_i)^x\right)^{\frac{1}{x}}$$

③ 对象 p 与簇 C 间的距离 $d(p,C)$ 定义为 p 与簇 C 的摘要之间的距离

$$d(p,C)=\left(\sum_{i=1}^{m}\mathrm{dif}(p_i,q_i)^x\right)^{\frac{1}{x}}$$

其中：$\mathrm{dif}(p_i,C)$ 为 p 与 C 在属性 D_i 上的距离，对于分类属性 D_i 其定义为 p 与 C 中每个对象在属性 D_i 上的距离的算术平均值，即 $\mathrm{dif}(p_i,C_i)=1-\dfrac{\mathrm{Freq}_{C|D}(p_i)}{|C|}$；对于数值属性 D_i 其定义为

$$\mathrm{dif}(p_i,C_i)=|p_i-c_i|$$

式中：$\mathrm{Freq}_{C|D}(p_i)$ 为频度，其定义为 C 在 D_i 上的投影中 p_i 的次数；摘要是由分类属性中不同取值的频度信息和数值型属性的均值两部分构成：

$$Summary=\{<Stat_i,Cen>|\,Stat_i=\{(a,\mathrm{Freq}_{C|D_i}(a))\,|\,a\in D_i\},1\leqslant i\leqslant m_C,Cen=(c_{m_{C+1}},\cdots,c_{m_C+m_N})\}$$

④ 簇 C_1 与 C_2 间的距离 $d(C_1,C_2)$定义为两个簇的摘要之间的距离

$$d(C_1,C_2)=(\sum_{i=1}^{m}\mathrm{dif}(C_i^{(1)},C_i^{(2)})^x)^{\frac{1}{x}}$$

其中：$\mathrm{dif}(C_i^{(1)},C_i^{(2)})$ 为 C_1 与 C_2 在属性 D_i 上的距离，对于分类属性 D_i 其定义为 C_1 中每个对象与 C_2 中每个对象的差异的平均值

$$\mathrm{dif}(C_i^{(1)},C_i^{(2)})=1-\frac{1}{|C_1|\bullet|C_2|}\sum_{p_j\in C_1}\mathrm{Freq}_{C_1|D_i}(p_j)\bullet\mathrm{Freq}_{C_2|D_i}(p_j)$$
$$=1-\frac{1}{|C_1|\bullet|C_2|}\sum_{q_j\in C_2}\mathrm{Freq}_{C_1|D_i}(q_j)\bullet\mathrm{Freq}_{C_2|D_i}(q_j)$$

对于数值型属性 D_i 其定义为

$$\mathrm{dif}(C_i^{(1)},C_i^{(2)})=|c_i^{(1)}-c_i^{(2)}|$$

例如假设描述学生的信息包含属性：性别、籍贯、年龄。有两条记录 p、q 及两个簇 C_1，C_2 的信息，可以求出记录和簇彼此间的距离为

p={男，广州，18}，q={女，深圳，20}

C_1={男：25，女：5；广州：20，深圳：6，韶关：4；19}

C_2={男：3，女：12；汕头：12，深圳：1，湛江：2；24}

按以上的定义，取 x=1 得到的距离如下：

$d(p,q)$=1+1+(20−18)=4

$d(p,C_1)$=(1−25/30)+(1−20/30)+(19−18)=1.5

$d(p,C_2)$=(1−3/15)+(1−0/15)+(24−18)=7.8

$d(q,C_1)$=(1−5/30)+(1−6/30)+(20−19)=2.63

$d(q,C_2)$=(1−12/15)+(1−1/15)+(24−19)=5.13

$d(C_1,C_2)$=1−(25*3+5*12)/(30*15)+1−6*1/(30*15)+24−19)=6.69

用以上的定义就可以使原来仅适用于数值属性或分类属性的聚类算法不受数据类型的限制而应用于任何数据类型。K-Summary 算法就是采用了以上定义的 K-means 算法，它由以下的主要步骤完成：

① 初始化：选择 K 个对象，创建 K 个簇的摘要信息(CSI)；

② 划分对象到最近的簇；

③ 重新计算每个簇的 CSI；

④ 重复步骤②和③直到选用的度量函数收敛，如误差和变化很小或相邻两次迭代没有对象从一个簇移动到另一个簇。

9.3 基于 MATLAB 的近邻法及 K-means 聚类法

例 2.55　胃病病人和非胃病人的生化指标测量值如表 9.1 所示。试用近邻法对某未知样本进行判别。

表 9.1　胃病病人和非胃病人生化指标的测定值

胃病类型	铜蓝蛋白（x_1）	蓝色反应(x_2)	吲哚乙酸(x_3)	中性硫化物(x_4)	归　类
胃病	228	134	20	11	1
	245	134	10	40	1
	200	167	12	27	1
	170	150	7	8	1
	100	167	20	14	1
非胃病	150	117	7	6	2
	120	133	10	26	2
	160	100	5	10	2
	185	115	5	19	2
	170	125	6	4	2
	165	142	5	3	2
	185	108	2	12	2
未知样本	225	125	7	14	
	100	117	7	2	
	130	100	6	12	

解：

在 MATLAB 中，有专门的 K-近邻法分类函数，其调用格式为

```
class =knnclassify (sample,training,group);
class =knnclassify(sample,training,group,k);
class = knnclassify(sample,training,group,k,distance);
class = knnclassify(sample,training,group,k,distance,rule);
```

其中 sample、training、group 分别为测试样本、训练样本及训练样本对应的类别号；K 为近邻法，默认值为 1；distance 为距离，可以选 euclidean、cityblock、cosine、Correlation、Hamming；rule 为表决规则，可以选 nearest、random、consensus。

```
>>load  x;  y=knnclassify(x(13:15,:),x(1:12,:),[0  0  0  0  0  1  1  1  1  1  1
1],2,'cityblock','nearest');
y=0  1  1
```

例 2.56　对例 2.55 的数据用 K-均值法进行聚类分析。

解：

在 MATLAB 中，有专门的 K-均值聚类算法函数，其调用格式为

```
IDX=kmeans(X,k)
[IDX,C]=kmeans(X,k)
[IDX,C,sumd]=kmeans(X,k)
[IDX,C,sumd,D] = kmeans(X,k)
```

```
    [...] = kmeans(...,param1,val1,param2,val2,...)
```

其中各参数的意义参见 MATLAB 中此函数的帮助文档。

还可以利用 silhouette 函数根据聚类结果绘制轮廓图。从轮廓图上能看出每个点的分类是否合理。轮廓图上第 i 个点的轮廓图定义为

$$S(i)=\frac{\min(i)-a}{\max[a,\min(b)]},i=1,2,\cdots,n$$

其中：$S(i)$为第 i 个点与同类的其他点之间的平均距离，$\boldsymbol{b}$ 为一个向量，其元素是第 i 个点与不同类的类内各点之间的平均距离。轮廓值 $S(i)$的取值范围为[−1,1]，$S(i)$值越大，说明第 i 个点的分类越合理，当 $S(i)<0$ 时，说明第 i 个点的分类不合理，还有比目前分类更合理的方案。

利用此函数对表中数据进行分析，可得到以下结果，可以看出，其中有两个样品的类别与原来的类别有所差异，而且如果函数用不同的参数进行计算，可得到不同的结果。

```
>>load x; y=kmeans(x,2,'distance','city')'
y=1  1  1  2  2  2  2  2  2  2  2  2  1  2  2
>> [s,h]=silhouette(x,y,'city');   %得图 9.1
```

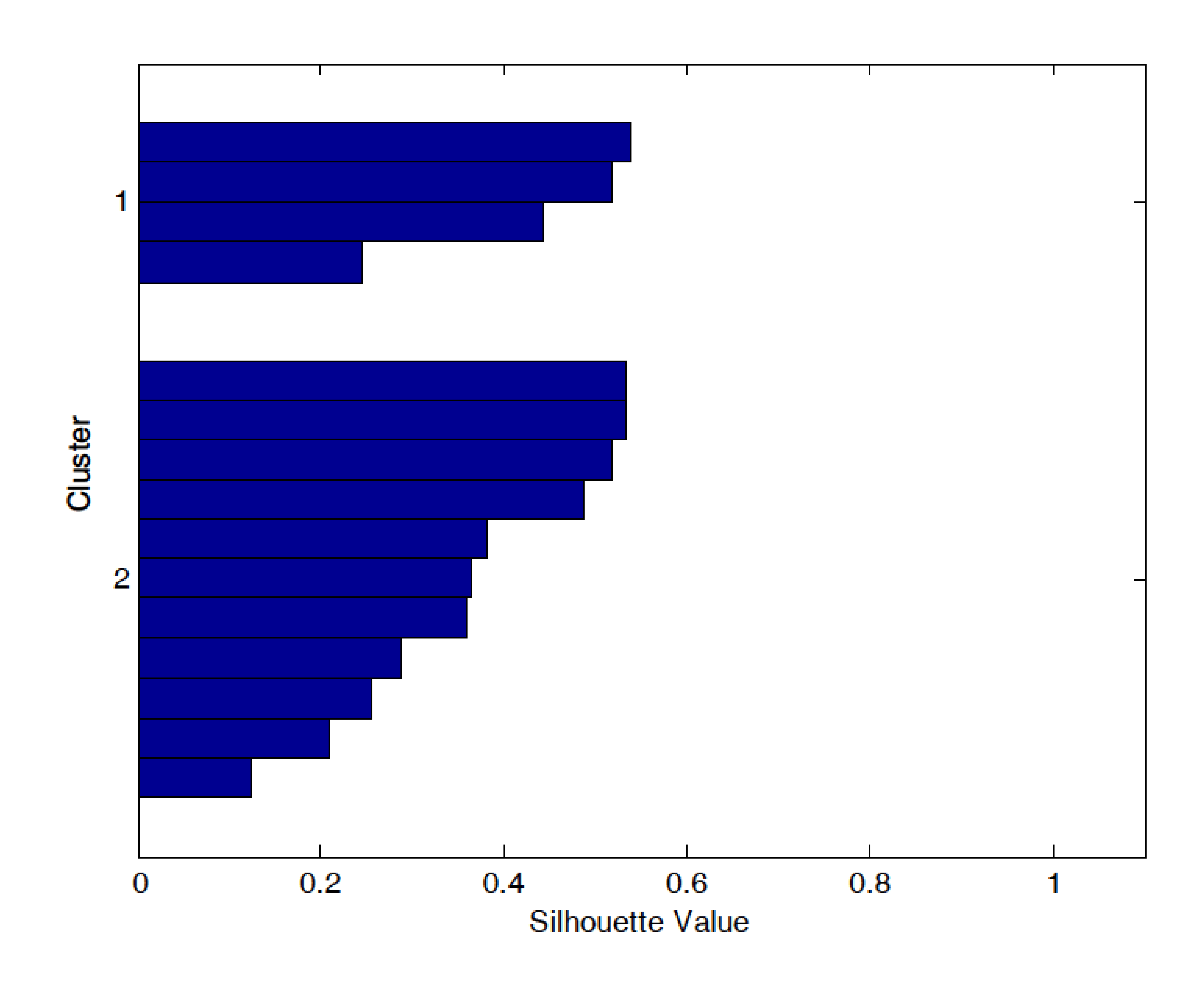


图 9.1　K-means 聚类值

例 2.57　对表 9.2 中的数据集，采用 K-Summary 算法将其划分为两个类。

表 9.2　某银行拖欠贷款情况数据表

序　号	是否有房	婚姻情况	年 收 入	拖欠贷款
1	yes	single	125K	no
2	no	married	100K	no
3	no	single	70K	no

续表

序　号	是否有房	婚姻情况	年 收 入	拖欠贷款
4	yes	married	120K	no
5	no	divorced	95K	yes
6	no	married	60K	no
7	yes	divorced	220K	no
8	no	single	85K	yes
9	no	married	75K	no
10	no	single	90K	yes

解：

根据 K-Summary 算法原理，编程计算如下，其中因为聚类是一种无指导的方法，所以计算时不使用标志位，即表中的数据只选用前三栏。

```
>>data={'yes'  'single' 125;'no' 'married'100;'no' 'single' 70;'yes' 'married'
       120;'no' 'divorced' 95;'no' 'married'  60;'yes' 'divorced' 220;'no'
       'single' 85;'no' 'married' 75;'no' 'single'90};
>>k=2;type=[1 1 0];
>> [y,ceter]=ksummary(data,k,type);   %y 为分类结果，ceter 为相应的分类中心
>> y{1}=2  3  5  6  8  9  10;
>> y{2}=1  4  7;
>> ceter{1}=total: 7
         proper: {{1x2 cell}  {1x3 cell}  [82.1429]}
>> ceter{2}=total: 3
         proper: {{1x2 cell}  {1x3 cell}  [155]}
```

读书笔记

第 3 篇　　数据挖掘相关技术

数据挖掘方法是由人工智能、机器学习的方法发展而来，结合传统的统计分析方法、模糊数学方法以及科学计算可视化技术，以数据仓库为研究对象，形成的数据挖掘的方法和技术。

第 10 章

数据仓库

10.1 概述

如何有效地管理企业在经营过程中所产生或收集的大量数据与信息，一直是信息管理人员所面临的一个重要问题。20 世纪 70 年代所出现的关系数据库在收集、存储、处理数据中发挥了重要的作用。随着市场竞争的加剧，信息系统的用户已经不能满足仅用计算机去处理日复一日的事务数据，而是需要信息即能够支持决策的信息去帮助管理决策。这就需要一种能够将日常业务处理中所收集到的各种数据转变为具有商业价值信息的技术。而传统数据库系统已经无法承担这一责任。

传统数据库对日常事务处理十分理想，但是要基于事务处理的数据库帮助决策分析，就产生了很大的困难。其原因主要是传统数据库的处理方式和决策分析中的数据需求不相称，导致传统数据库无法支持决策分析活动。这些不对称主要体现在决策处理中的系统响应问题、决策数据需求的问题和决策数据操作问题。

为了解决传统数据库存在的这些问题而将其用于决策系统，通过数据库发展而衍生出数据仓库。

数据仓库是近年来在信息管理领域得到迅速发展的一种面向主题的、集成的、随时间变化的、非易失性数据的集合，其目的在于支持管理层的决策。数据仓库通常主要包含数据仓库数据库、数据集市知识挖掘库、数据源、数据准备区以及各种管理工具和服务工具。数据仓库建立后，先要从数据源中抽取所需的数据到数据准备区，在数据准备区中经过数据的净化处理，再加载到数据仓库数据库中，最后根据用户的需求将数据发布到数据集市中。当用户使用数据仓库时，可以通过联机处理（OLAP）、数据挖掘等数据仓库应用工具，向数据集市知识挖掘库或数据仓库进行决策查询分析或知识挖掘。所以从本质上讲，数据仓库是数据库技术的一种新的应用，是一种解决方案，能够对原始的操作数据进行各种处理并转换成有用信息，用户可以通过分析这些信息做出策略性决策。到目前为止，大多数数据仓库产品还是用数据库管理系统来管理其中的数据。

数据仓库虽然是以数据库发展而来的，但是两者在许多方面都存在着相当大的差异，如表 10.1 所示。

表 10.1　数据仓库与数据库对比表

对比内容	数 据 库	数据仓库
数据内容	当前值	历史的、存档的、归纳的、计算的数据
数据目标	面向业务操作程序，重复处理	面向主题域，分析应用
数据特性	动态变化、按字段更新	静态、不能直接更新，只能定时添加、刷新
数据结构	高度结构化、复杂，适合操作计算	简单、适合分析
使用频率	高	中到低
数据访问量	每个事务只访问少量记录	有的事务可能需要访问大量记录
对响应时间的要求	以秒为单位计算	以秒、分甚至时为计算单位

10.1.1　数据仓库重要特性

数据仓库可以定义为：一个面向主题的、集成的随时间变化的非易失性数据的集合，用于支持管理层的决策过程。从定义中可以发现数据仓库具有这样一些重要的特性：面向主题性、数据集成性、数据的时变性、数据的非易失性、数据的集合性和支持决策作用。

1. 面向主题性

面向主题性表示数据仓库中数据组织的基本原则，数据仓库中的所有数据都是围绕着某一主题组织、展开的。所谓的主题就是在一个较高的管理层次上对信息系统中的数据按照某一具体的管理对象进行综合、归类所形成的分析对象，也可以是一些数据集合，这些数据集合对分析对象进行了比较完整的、一致的数据描述，这种描述不仅涉及数据自身，还涉及数据之间的数据。在主题的划分中，必须保证每个主题的独立性，也就是说，每一个主题要具有独立的内涵，明确的界线。确定主题以后，需要确定应该包含的数据。此时应注意不能将围绕主题的数据与业务处理系统中的数据相混淆。

如在企业销售管理中的管理人员所关心的是本企业哪些产品销售量大、利润高，哪些客户采购的产品数量大，竞争对手的哪些产品对本企业产品构成威胁，根据这些管理决策的分析对象，就可以抽象出“产品”“客户”等主题。但诸如“产品订单”“产品库存”等有关“产品”的数据只是业务处理系统中的业务操作数据，并不能完成对“客户”的分析，因为还缺少客户的产品采购量、最后一次采购时间、购买竞争对手的产品等数据。所以在确定“客户”这一主题，需要重新进行数据的组织。

2. 数据集成性

数据仓库的集成性是指根据决策分析的要求，将分散于各处的源数据进行抽取、筛选、清理、综合等集成工作，使数据仓库中的数据具有集成性。

数据仓库所需要的数据并不是直接从业务发生地获取，而是从与业务处理发生直接联系的业务处理系统中获取，因此需要对数据进行一系列的预处理，即数据的抽取筛选、清理和综合等集成工作，将数据源中数据的单位、字长与内容统一，消除源数据中字段的同名异义、异名同义等现象。

3. 数据的时变性

数据的时变性是指数据应随着时间的推移而发生变化。数据仓库的数据不能长期不变，必须能够不断地捕捉业务系统中的变化数据，将那些变化的数据追加到数据仓库中去，即不断地生成业务数据库的快照，以满足决策分析的需要。这些快照可以产生数据仓库的连续动态变化图，有助于决策分析。

数据仓库数据的变化，不仅反映在数据的追加方面，而且还反映在数据删除上。数据仓库中数据的存放期一般为 5~10 年，越过此时间则删除。

数据仓库中数据的变化还表现在概括数据的变化上。数据仓库中的概括数据是与时间有关的，概括数据需要按照时间进行综合、按照时间进行抽取。为满足数据仓库中数据的时变性需要而进行的操作称为数据刷新。

4. 数据的非易失性

数据仓库的非易失性是指数据仓库中的数据不经常进行更新处理，因为数据库中的数据大多表示过去某一时刻的数据，主要用于查询。数据的非易失性可以支持不同的用户在不同的时间查

询相同的问题时，获得相同的结果。

5. 数据的集合性

数据仓库的集合性意味着数据仓库必须以某种数据集合的形式存储起来。目前数据的集合方式主要是以多维数据库方式进行存储的多维模式，以关系数据库方式进行存储的关系模式或以两者相结合的方式进行存储的混合模式。

6. 支持决策作用

数据仓库组织的根本目的在于对决策的支持。管理决策者可从貌似平淡的数据中敏锐地发现众多的商机，为决策者对数据的自我分析提供了便利，提供了辅助决策分析的有力工具。

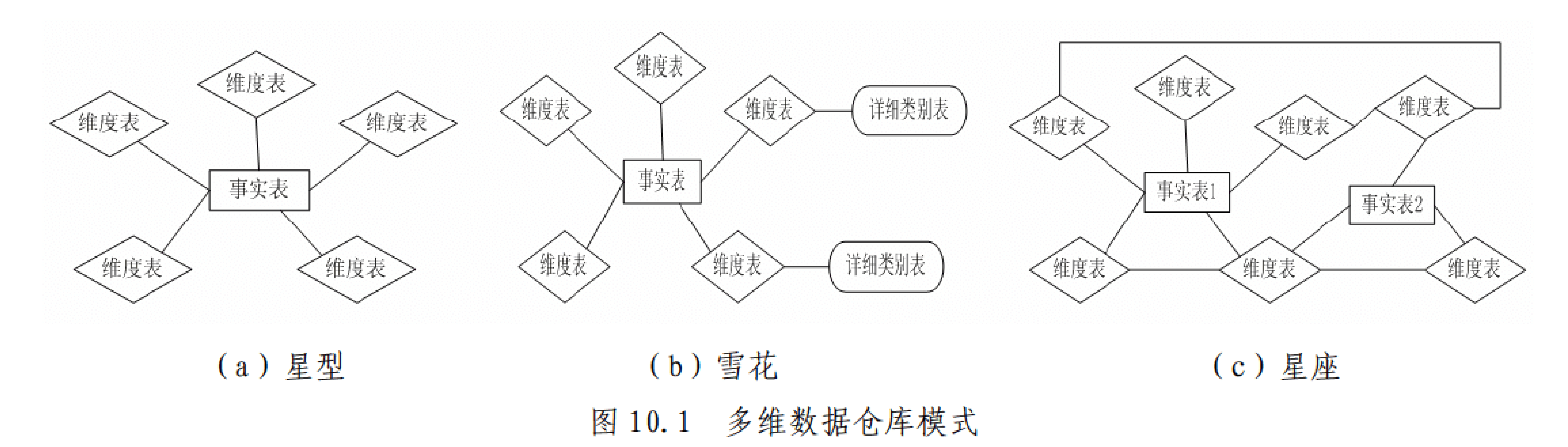
10.1.2 数据仓库中几个重要概念

1. 维

在应用数据仓库进行决策分析时，经常需要选择一个对决策活动有重要影响的因素去进行决策分析。因此，用户在使用数据仓库时所使用的决策分析角度或决策分析出发点构成了数据仓库的维。如客户、产品或供应商、地点、渠道、事件发生的时间等角度都可以是数据仓库的维。

数据仓库的维还可以作为数据仓库操作过程的途径，这些路径通常位于维的不同层次结构中，例如客户可以按地理位置进行分组：街道、县、市、省。这样就可以按街道、县、市、省的先后次序进行数据的“上卷”和“下钻”。前者是指用户在数据仓库的应用中，从较低层次的数据开始逐步将数据按层次进行概括处理；后者是指从数据仓库中的高层数据开始逐步向低层数据探索，了解概括数据的具体细节。

现在最流行的数据仓库多为多维数据模型，可分为星型、雪花和星座三种模式，如图 10.1 所示。



（a）星型　　（b）雪花　　（c）星座

图 10.1　多维数据仓库模式

2. 数据立方体

当用户观察某一事务的角度不同时，围绕该事务会产生多个观察角度，也即产生多维。数据仓库中的多种维交点，就是数据仓库用户所需要观察的事务。数据仓库的立方体实际上是一个包含用户需要观察数据的集合体，它提供企业感兴趣的商业事务。如图 10.2 即为数据立方体。

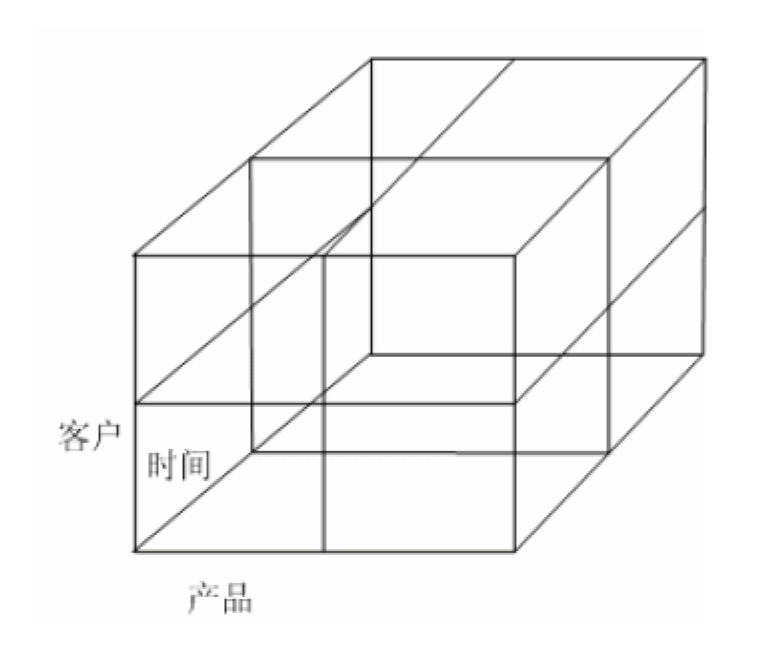


图 10.2　数据立方体

3. 聚集

聚集或聚合是指收集了基本事务数据的结构。在一个立方体中包括很多层次，这些层次可以向用户提供某一层次的概括数据。例如地区销售经理想了解本地区的销售总量、未来的销售趋势、客户的类型，就需要按本地区的城市、街道、产品种类和客户类型进行概括，也就是进行聚集。通过聚集，形成基于维的有决策分析意义的一些数据交集。

4. 数据颗粒度

数据颗粒度是指数据的细化程度。数据粒度越小，信息越细，数据量越大；颗粒粒度越大，就忽略了众多的细节，数据量越小。

数据的综合程度还会影响数据的用途。对于非常细致的问题，细节数据非常合适，但对于综合程度较高的问题，使用综合数据就可以迅速回答这个问题。

粒度的另一种形式是抽样率，即以一定的抽样率对数据仓库中的数据进行抽样后得到一个样本数据库，数据挖掘在这个样本数据库上进行。

5. 元数据

元数据是指数据仓库创建过程中产生的有关数据源定义、目标定义、转换规则等关键数据，是定义数据仓库对象的数据。元数据还包含关于数据含义的商业信息，如图 10.3 所示。

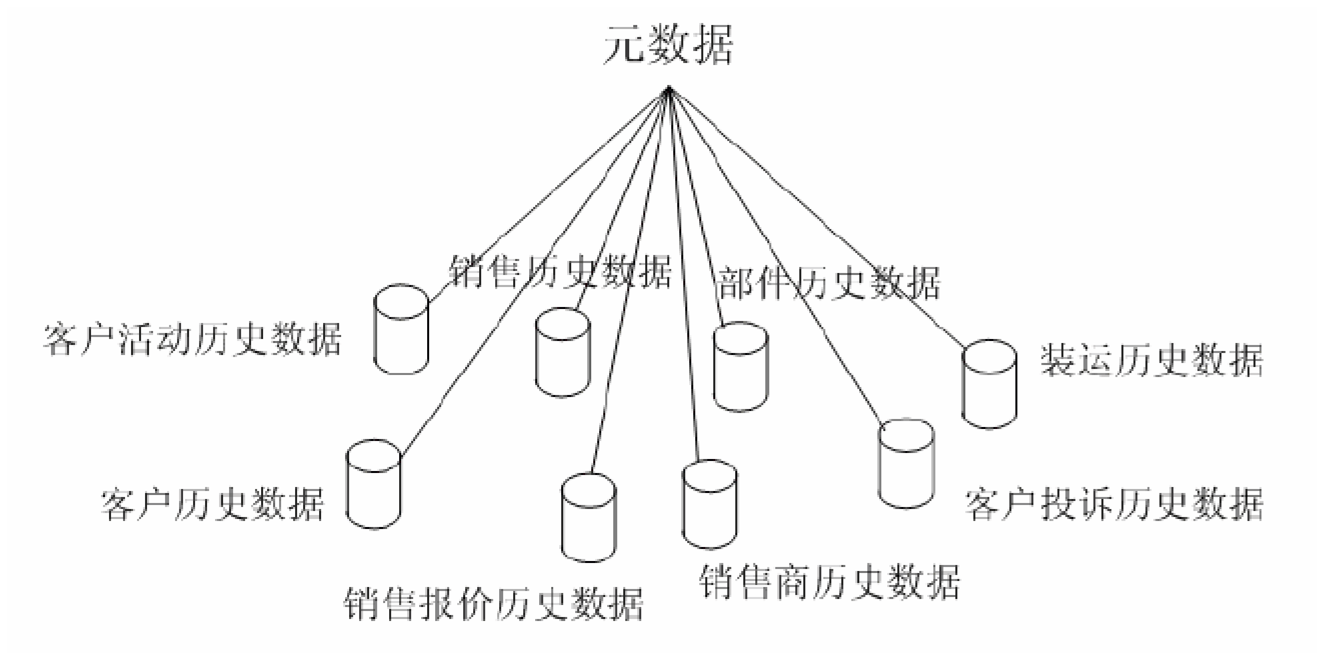


图 10.3　元数据

元数据用作目录，可以帮助决策支持系统者定位数据仓库的内容；当从操作环境转移到数据仓库环境时，元数据可以作为数据映射指南。

10.2 数据仓库设计

下面以图 10.4 所示的某中药数据仓库的设计为例，介绍数据仓库设计的一些基本概念。

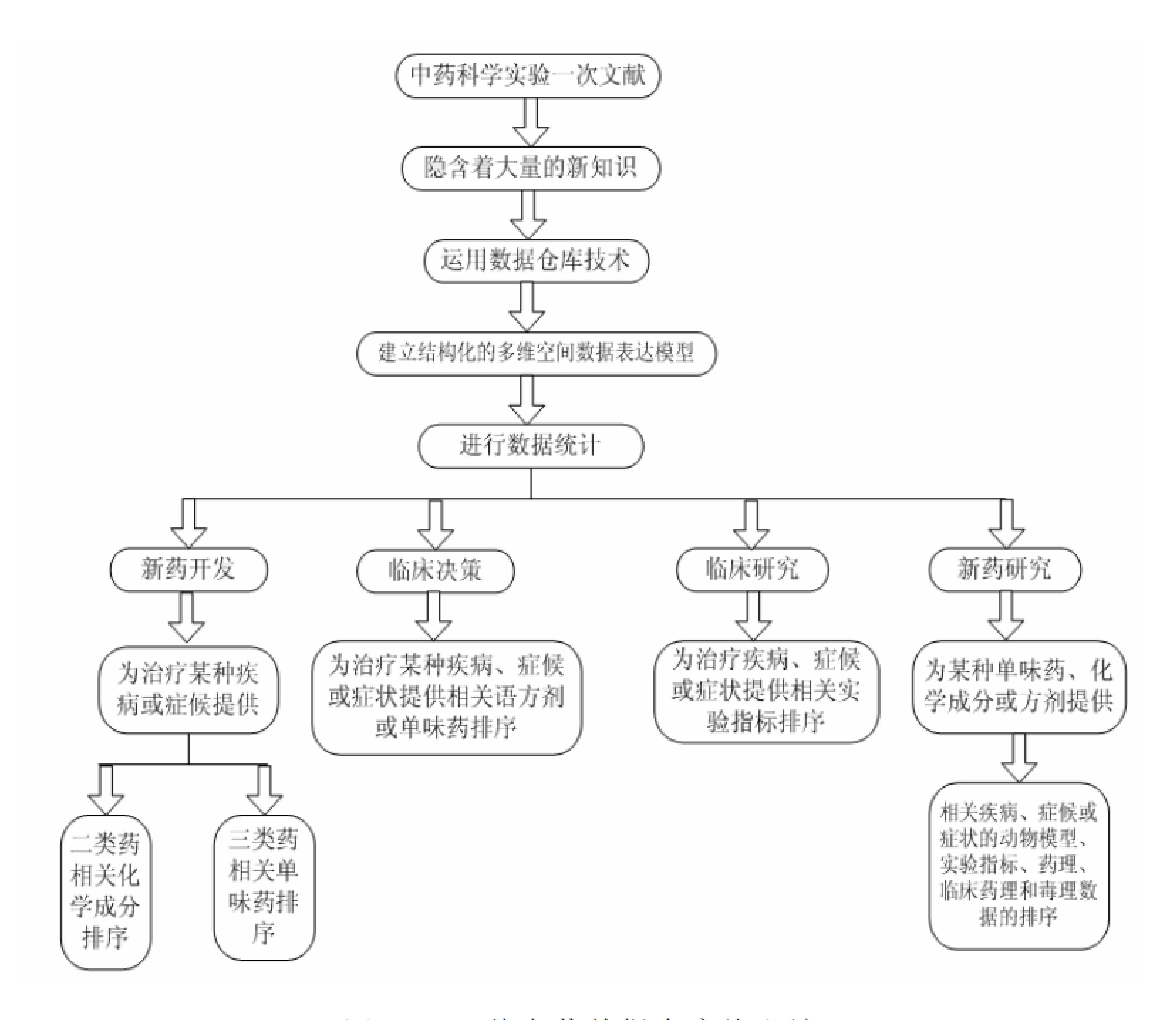


图 10.4　某中药数据仓库的设计

10.2.1 数据仓库的总体结构

数据仓库是近年 IT 技术和信息管理迅速发展的结果。如果从数据仓库的概念结构看，应该包括数据源、数据准备区、数据仓库数据库、数据集市知识挖掘库以及各种管理工具和应用工具，结构框图如图 10.5 所示。

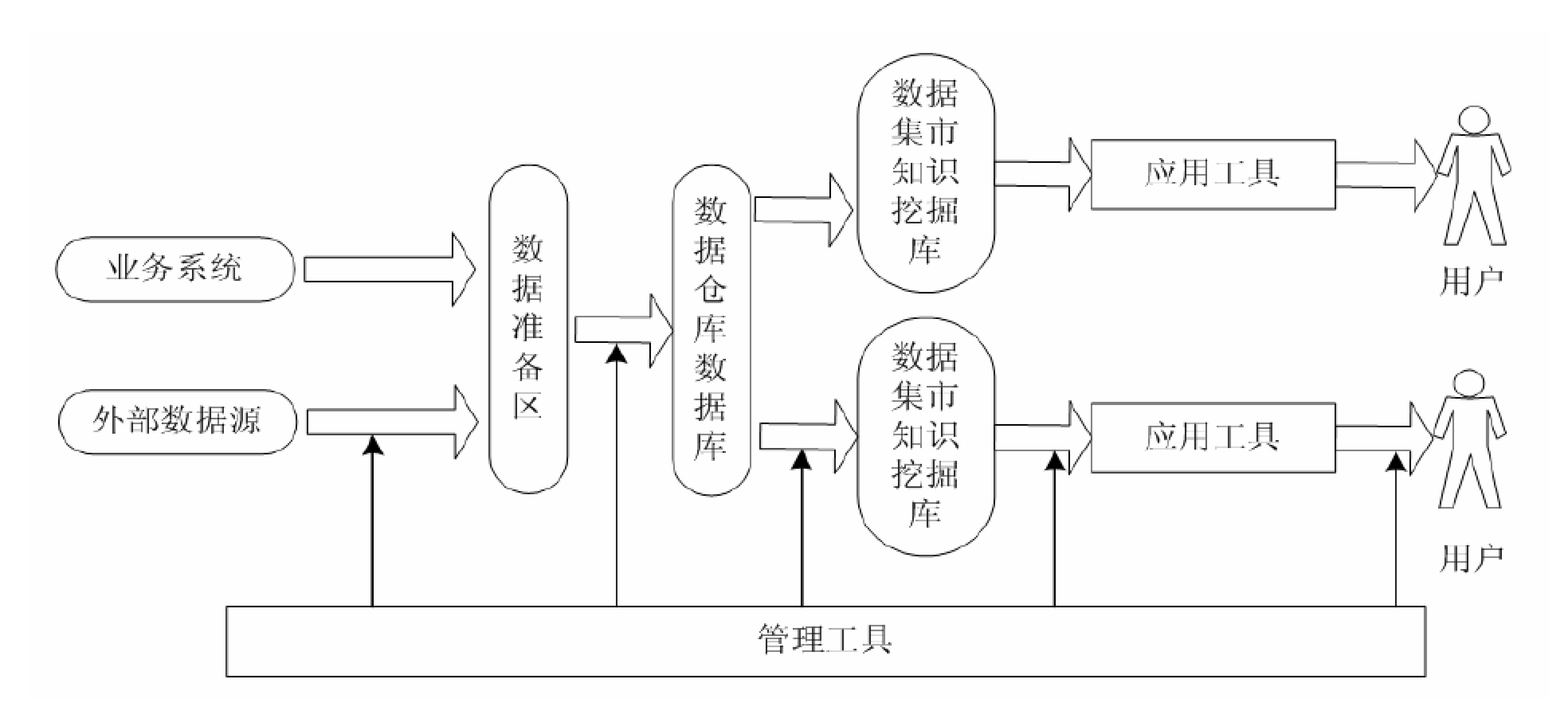


图 10.5　数据仓库的概念仓库

10.2.2　数据仓库的基本功能层

数据仓库的基本功能部分包含数据源、数据准备区、数据仓库数据库、数据集市知识挖掘库，以及存取与使用功能部分。

（1）数据源。数据源是指存储在数据仓库中的数据来源，它包含业务数据、历史数据、办公数据、Web 数据、外部数据以及数据源元数据。

在这些数据加载到数据仓库中前，应使其格式符合数据仓库中数据的格式，加载到数据仓库中的数据应具有重要的使用价值。

（2）数据准备区。由于数据仓库的数据来源十分复杂，这些数据在进入数据库之前常常需要在数据准备区进行筛选、清理等标准化处理。因此数据准备区由数据标准化处理、数据的过滤与匹配、数据的净化处理、标明数据的时间戳、确认数据质量与元数据抽取和创建等操作组成。

数据的标准化处理主要是将同名不同内容的、同内容而不同名的、同名同内容但不同结构的数据进行标准化处理，以便在数据仓库的使用中不至于产生混乱。

数据的过滤与匹配主要是对进入数据仓库的数据按照用户的需要进行筛选，将用户不需要的数据从数据源中删除，而留下的数据要能够与数据仓库用户的需求相匹配。

数据的净化处理主要是对准备加载到数据仓库中的数据进行正确性判断，将那些数据内容错误、格式错误或类型错误的数据进行修正、净化处理。

（3）数据仓库数据库。数据仓库数据库由数据重整、数据仓库创建以及元数据管理部分组成。

数据重整是为使数据仓库能够更好地为用户服务所进行的一系列预操作，包含数据的集成与分解、数据的概括与聚集、数据的预算与推导、数据的翻译与格式化、数据的转换和元数据创建。

数据的集成是指对来自不同系统中的数据进行集成，以创建新的数据。有时还要按照数据库处理的需要将一个表中的数据分解成数据仓库中的两个或多个数据块。

数据仓库在存储数据时，经常按数据的时间顺序、业务范围、发生地域等进行分割存储，以便于用户的分析和提高数据仓库的使用效率，但是，在实际操作中又经常需要对数据进行概括与聚集处理，即根据某一属性对数据进行汇总。

为了提高数据仓库的使用效率，在数据仓库中需要事先对使用者的常规操作进行预先设置，即无须用户干预就可实现数据的一些计算即为数据的预算与推导。

数据的翻译与格式化是指对来自不同数据源的数据进行翻译和格式化处理，便于今后的统一处理。

数据的转换是因为数据仓库的数据源中的数据结构基本上是标准关系模式，而数据仓库则大多数采用星型或雪花模型，这两者的差异必须依靠数据的转移与映像来消除，也就是将这两者不同的数据模式以某种方式连接起来，将数据源数据转化为适合数据仓库事实表的行的过程。

（4）数据仓库的创建是指完成数据仓库的建模、数据的概括、数据的聚集、数据的调整与确认、建立结构化查询和创建词汇表。

数据仓库的建模是指或从已经创建的数据模型中导出数据仓库的数据模型，或构造新的数据模型。在数据仓库模型的设计过程中，需要完成数据的分割、主题域和粒度的确认。

数据概括是指根据用户的需要对数据进行概括，从初步的概括数据中创建用户所需要的高度

概括数据。

在数据仓库中，常要根据一些典型的查询要求，对数据仓库中数据聚集处理，以提高效率。例如可以对产品的地区分布、品牌的分布进行事先聚集，才能使用户在数据仓库的使用中每次都感受到信息使用时间的一致性。

在数据完成概括聚集以后，需要对概括与聚集后的数据进行确认，如果数据概括、聚集的效果不好，还需要进行一些调整，以保证数据仓库的使用效果。

建立结构化查询是指为了提高一些结构化查询，可以预定义这些查询，且将这些结构化查询作为元数据存储在元数据库中。当用户进行数据仓库的实际查询应用时，只要从元数据库中取回，可以大大提高数据仓库的运行效率。

在创建数据仓库的过程中，需要根据所捕获的元数据建立元数据的词汇表。在词汇表中一般需要包含元数据的名称、别名、简述、创建时间、上次更新时间、关键词、数据来源、转移/转换信息、概括或推导算法等内容。

（5）数据集市知识挖掘库结构。数据集市知识挖掘库的功能与数据仓库的功能极为相似，只是数据集市的目的是为某个部门或某个领域的用户提供服务，而数据仓库则是为全体用户提供服务。因此，可以将数据集市知识挖掘库看成数据仓库的一个逻辑上或物理上的子集。数据集市知识挖掘库中也包含用户所需要查询的详细数据和概括性数据。

（6）数据仓库的数据存取与使用。数据仓库的数据存取与使用是使用数据仓库的最终目的，为数据仓库的最终用户提供决策分析和挖掘知识的功能。它可分为数据仓库存取与检索部分以及数据仓库分析与报告部分这两部分。

数据仓库存取与检索部分为用户提供访问数据仓库或数据集市的功能，利用这些功能可将用户检索的数据转换成多维数据并且存入多维数据库，可以将数据仓库或数据集市中的数据“卸载”下来，成为局部存储数据，便于用户进行局部分析、数据查询、翻译转换等处理。

为了用户使用方便，还应提供管理与使用数据仓库元数据功能。这些功能可以帮助用户了解数据仓库或数据集市的名称、描述说明、数值、价值来源以及版本等内容，了解数据的名称、数据等内容和数据从抽取到存入数据仓库或数据集市的转移过程，了解数据的定位和数据的可靠性，以及如何存取和使用数据。利用这些功能可以帮助用户掌握数据的正确内容、信息的粒度、信息的概括程度、原始数据的来源和日期，并且可以按其上下文查看数据，将数据转化为信息。

数据仓库分析与报告为最终用户使用数据仓库提供一组工具，可使用户依靠数据仓库或数据集市进行决策分析或知识挖掘。这些工具包括报表处理工具，分析与决策支持工具、业务建模与分析处理工具，数据挖掘工具等。如地理信息系统（GIS），数据采集工具、联机分析处理、可视化工具、统计工具、浏览器、图形用户界面建立程度、电子表格、报表生成器和数据访问工具等。

（7）数据仓库的管理层。数据仓库管理层由数据仓库的数据管理和数据仓库的元数据管理组成。

数据仓库的数据管理层包括数据抽取、新数据需求与查询管理，数据加载，存储、刷新和更新系统，安全性与用户授权管理系统以及数据归档、恢复及净化系统等部分。

10.2.3 数据仓库技术

尽管在许多情况下，数据仓库的创建与使用技术并不比数据库创建使用的技术复杂，但是数

据仓库的创建与使用技术也有许多特定要求。

（1）数据管理技术。数据管理技术包含大批量数据管理技术、数据仓库索引与数据监视技术、元数据管理技术、数据压缩技术和复合键码技术。

（2）数据存储技术。数据的存储技术包含多介质存储设备的管理技术、数据存储的挖掘技术、数据的并行存储与管理技术、可变长技术和锁切换技术。

（3）数据仓库接口技术。数据仓库的接口技术包含多技术接口技术、语言接口技术和数据的高效率加载技术。

10.2.4　数据仓库设计

对于一个企业或组织而言，建立数据仓库是一个巨大和长期的工作。由于公司战略可能会在数据仓库开发期间发生改变，从而对公司而言存在大量的未知因素和风险，所以在数据仓库开发阶段选择适当的方法可以降低这些风险和未知因素。

建造数据仓库有两个主要部分，即与操作系统接口的设计和数据仓库本身的设计。数据仓库系统设计的目标是建立一个全局一致的数据环境，以此作为企业决策支持系统的基础。它的开发是从最基本的主题开始，不断地发展新的主题，完善已有的主题，最终建立起一个面向主题的分析型数据环境，另外，在这个过程中，用户的需求是模糊的，这就决定了不可能从用户需求出发进行数据仓库的设计。数据仓库设计的这些特点就决定了在设计过程中要采用"数据驱动"的系统设计方法。所谓的"数据驱动"设计方法是以数据为基础，进行从面向应用到面向分析需求的转变，并逐步提高决策效果的方法。

数据仓库系统开发时，有两种基本的策略可供选择：第一种是自顶向下的策略。先建立一个全局数据仓库的结构，然后在此基础上建立部门的数据集市和个人的数据仓库。这是一种系统解决方案。第二种是自底向上的策略。就是数据集市方法，它可以从最关心的部分开始，先以最少的投资，完成企业当前的需求，然后再不断放弃、完善。

10.2.5　数据仓库设计步骤

在数据仓库设计过程中，需要建立三个层次的模型：①概念模型；②逻辑模型；③物理模型。这三个模型与现实的变化联系，可用图 10.6 表示。

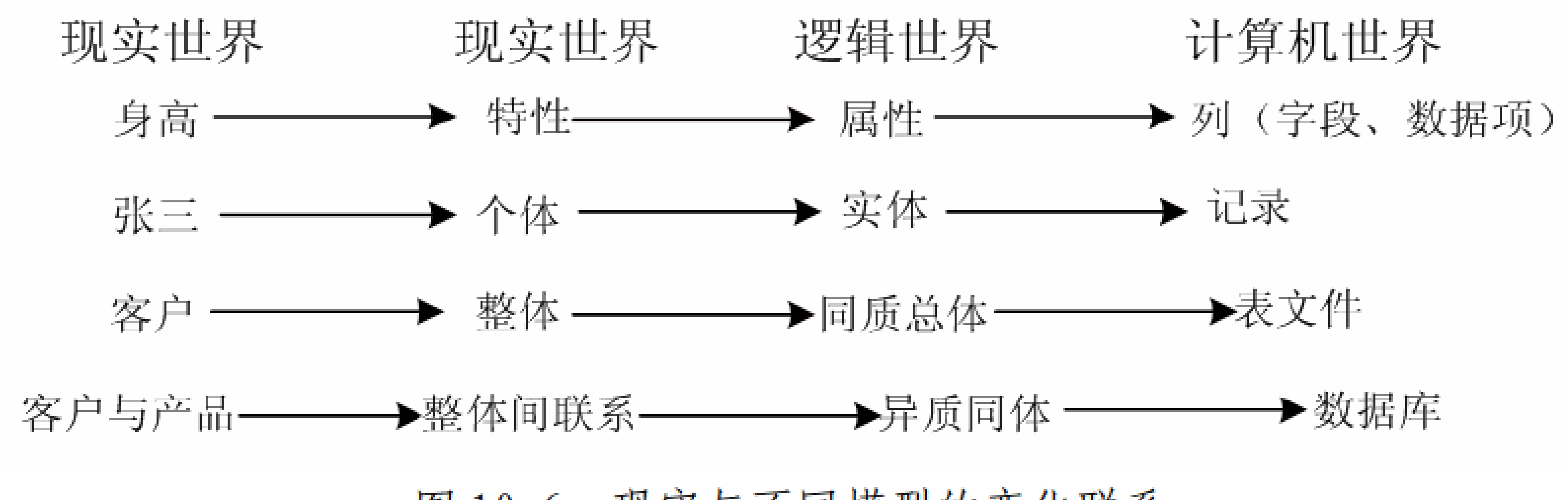


图 10.6　现实与不同模型的变化联系

1. 概念模型的设计

概念模型是联系主观与客观的桥梁，它是一个为一定的目标设计系统、收集信息而服务的概念性工具。在计算机系统设计中，概念模型的设计就是创建一种基于对象、代表实际业务的模型。

由于概念模型是面向现实的，所以在认识和设计系统时，概念模型应易于修改而且适应性强。

2. 逻辑模型的设计

数据仓库的逻辑模型应该与数据仓库物理实现时所使用的数据库有关，它主要是关系模型。

在进行数据仓库的逻辑设计中，一般需要完成分析主题域、确定装载到数据仓库的主题，确定粒度层次划分，确定数据分割策略、关系模式的定义和记录系统定义、确定数据抽取模型等。逻辑模型的最终设计成果应该包含每个主题逻辑定义，且将相关内容记录在数据仓库的元数据中，其中包括粒度划分、数据分割策略、表划分和数据来源等。

3. 物理模型的设计

数据仓库的物理模型就是逻辑模型在数据仓库中的实现模式。其中包括逻辑模型中各种实体表的具体化。例如表的数据结构类型、索引策略、数据存放位置以及数据存储分配等。在进行物理模型设计实现时，所考虑的因素有 I/O 存取时间、空间利用率和维护的代价。

为了确定数据仓库的物理模型，设计人员必须要做到以下几方面的工作：首先要全面了解所选用的数据库管理系统，特别是存储结构和存取方法；其次，了解数据环境、数据的使用率、使用方式、数据规模以及响应时间要求等，这些都是对时间和空间效率进行平衡和优化的重要依据；最后，还要了解外部存储设备的特征。只有这样才能在数据的存储需求与外部存储设备条件中获得平衡。

10.3 数据仓库的开发应用

数据仓库的开发可分三个阶段：数据仓库规划分析阶段、数据仓库设计实施阶段以及数据仓库的应用。这三个阶段不是简单的循环往复，而是不断完善、提高的过程。一方面通过这三个阶段的数据仓库开发，积累了数据仓库的开发应用经验，可以转向其他主题的数据仓库应用；另一方面通过数据对原始数据仓库的开发应用经验积累，可对原始数据仓库提出改进的建议，使原始数据仓库通过改进得到提高，如图 10.7 所示。

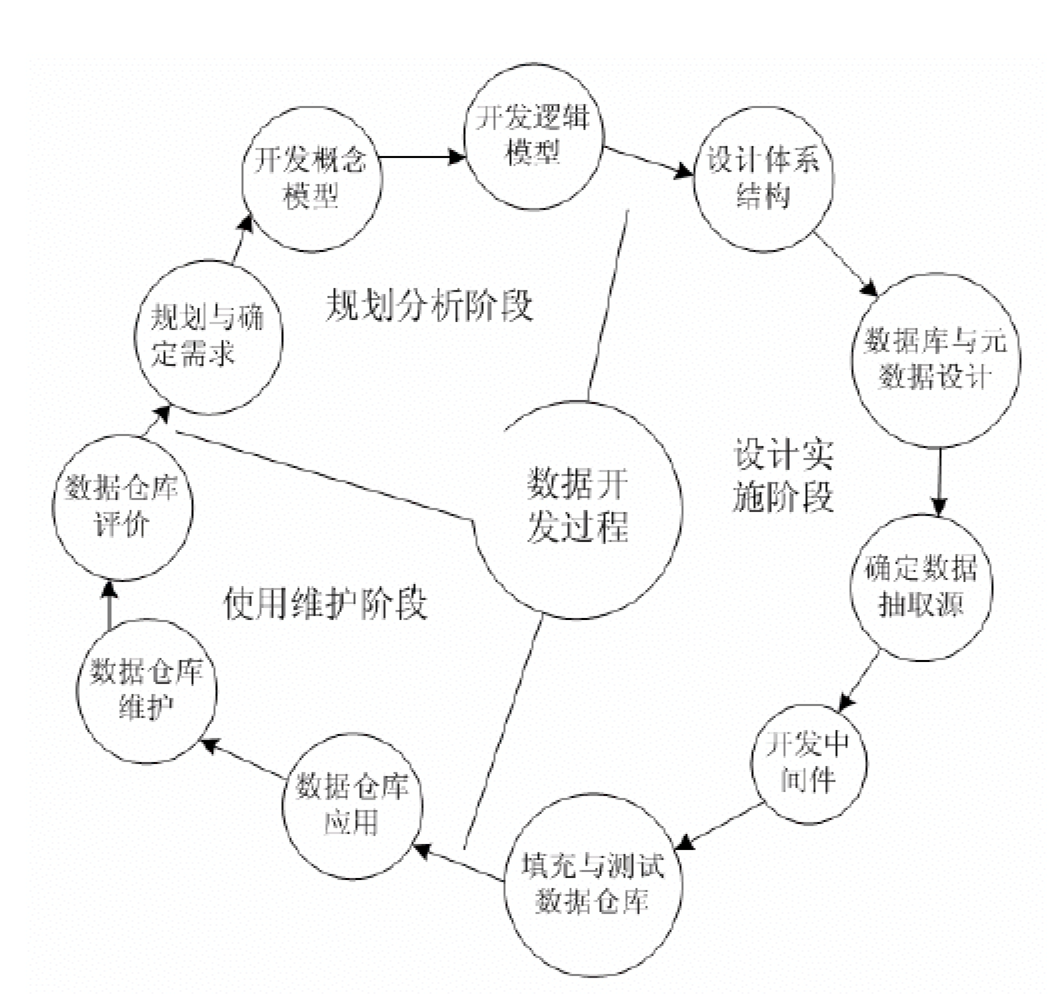


图 10.7 数据仓库生命周期开发应用全过程

数据仓库规划分析阶段的工作内容主要包括：调查、分析数据仓库环境，完成数据仓库的开发规划，确定数据仓库开发需求；建立包括实体关系图、星型模式、雪花形式、元数据模型以及数据源分析的主题区数据模型，并且根据主题区模型开发模型数据仓库逻辑模型。

数据仓库设计实施阶段的工作内容主要包括：根据数据仓库的逻辑模型设计数据仓库体系；设计数据仓库与物理数据库；用物理数据库元数据填充面向最终用户的元数据库；为数据仓库中每个目标字段确定它在业务系统或外部数据源中数据来源；开发或购买用于抽取、变换和合并数

据等中间件的程序；将数据从现有系统中到仓库中，填充数据仓库且测试。

数据仓库的使用维护阶段的工作内容主要包括：数据仓库的投入使用，且在使用中改进、维护数据仓库；对数据仓库进行评价，为下一个循环开发提供依据。

下面以某个超市的数据仓库设计为例，简要地说明数据仓库设计开发的一般过程。

日趋激烈的市场竞争要求超市经营者更加准确地了解超市经营状况，跟踪市场趋势，更加合理地制定商品的采购与销售策略。由于超市业务处理的需要，一般在人事、采购、库存、销售等部门有着人事、采购、库存、销售的数据库，分别处理各自的业务。但是各个部门的数据库都是按各部门的业务需要加以组织的，缺乏全局性，超市管理层决定要在这些数据库的基础上建立一个数据仓库。

10.3.1　数据仓库概念模型设计与开发

概念模型的设计可以分为用户的需求调查、模型定义、模型分析和模型设计几个阶段。

1. 概念模型的需求调查

当用户需要开发一个数据仓库时，往往提出一个数据仓库开发的任务书。在任务书中对组织的背景和组织所在行业的发展进行必要的论述，说明组织目前所要完成的业务功能以及业务范围，且就行业的发展状态，提出组织的战略发展目标，然后，就实现这一发展战略需要数据仓库在决策方面提供哪些支持。

开发的超市数据仓库任务书的内容有：数据仓库用于支持对存在激烈市场竞争的零售行业分析，数据仓库能向管理部门提供关于客户、客户购买行为，以及国内外零售行业的市场信息。

为完成这一数据仓库的开发任务，数据仓库开发者首先要向有关人员和部门进行调查，描绘关于这一数据仓库以及数据仓库所在环境的完整画面。调查范围需要从组织中负责数据仓库开发的项目负责人开始，而后扩展到知识用户、信息用户和信息管理人员。调查时要注意不应向调查人员询问数据仓库应该具有什么功能，而是从管理决策工作中关于数据的需求问题，用户基本情况，用户使用信息的情况、对数据仓库的看法和评价等角度进行调查。

用户的信息要求可以从项目负责人的调查中得到，在此过程中需了解：用户对系统的希望和要求；哪些事务或业务与任务说明书中的业务需要相关，与这些事务或业务有关的数据保存在哪些系统中？管理人员在进行决策分析时，一般需要多长时间的数据？现在组织中使用的业务处理系统是否能够提供这些决策分析数据等内容。在与用户有关的调查中则需要了解：用户是哪些人，他们应该怎样与数据仓库发生关系？这些用户是否拥有自己的计算机系统，在这些系统中配置了哪些信息处理系统，这些系统的环境如何？用户在工作中是否使用了数据分析工具，他们在分析工作中经常做哪些方面的分析，是市场的，还是金融的？用户在使用分析报告时喜欢静态文本方式的，还是动态在线的？

在对知识用户与信息用户的调查中需要了解关于信息的来源：用户在组织中承担什么工作，在工作中所需要的信息，信息中是否有战略信息，这些信息的来源是哪里？这些信息采用哪些工具处理，在所在的部门中使用哪些信息系统，这些系统提供哪些分析信息，以及提供信息的方式。还要了解关于用户的一些基本情况，如用户的计算机系统环境、用户的知识状况等。

2. 概念模型的定义

在概念模型的定义过程中需要确定系统的范围以及所涉及的对象。为实现超市数据仓库概念模型的定义，首先需要分析用户的决策需求；其次，分析为实现这些决策分析，数据仓库应该提供哪些信息？具体到超市数据仓库而言，它的决策分析有：客户的购买趋势、商品供应市场的变化趋势，供应商和客户的信用等级等情况。为完成这些决策分析，需要商品销售量、商品采购量、商品库存量、客户情况和供应商等这样一些数据。

为了对数据进行完整的、规范的分析，可以采用用户信息需求表来描述用户的信息需求状况。在需求表中列出概述模型定义中所确定的数据仓库用户决策分析问题以及所需要的信息。在列出所有信息的同时，还要明确这些信息的详略程度。表 10.2 即为用户信息需求表，表中数字为对应概念的值。

表 10.2　用户信息需求表

决策分析问题	客户购买商品趋势分析					
需求信息表类	日　期	地　点	商　品	年 龄 组	经济状况	信　用
需求信息 1 层	年（4）	国家（15）	商品种类（7）	年龄组	经济类	信用（10）
需求信息 2 层	季（16）	省（60）	商品小类（40）	（8）	（10）	……
需求信息 3 层	月（48）	市（200）	商品（220）	……	……	
需求信息 4 层	……	街道（2100）	……			
需求信息 5 层		商店（20000）				
……		……				

概念模型的定义不仅需要构建一个企业数据模型 ER 图（ERD），即描述组织业务的蓝图，包括整个组织系统中各个部门的业务处理及其业务处理数据，如图 10.8 所示；还要了解 ERD 模型中每一个实体的诞生与消亡事件。如在销售业务处理系统中，某个客户第一次购买产品，系统会将一些相关信息记录在案，但是某个已经记录在案的客户，如果现年没有订购产品，就要在业务系统中将其置于停顿状态；如果某个客户三年没有订购产品，就要从其业务系统中删除，但在数据仓库中，该客户的信息必须长期保留，因为管理人员可能需要了解五年中的客户信息，数据仓库就需要提供销售情况的五年相关信息，如客户的第一次订购时间、最后一次订购时间、目前的状况等。为获取这些信息，在数据仓库的高层模型中就需要使用 CRUD 工具反映实体的生成、引用、更新和删除情况。CRUD 是指创建（Create）、读取（Read）、更新（Update）、删除（Delete）一个或多个数据项来互相连接每个应用程序。表 10.3 为实体与功能关系 CRUD 矩阵。

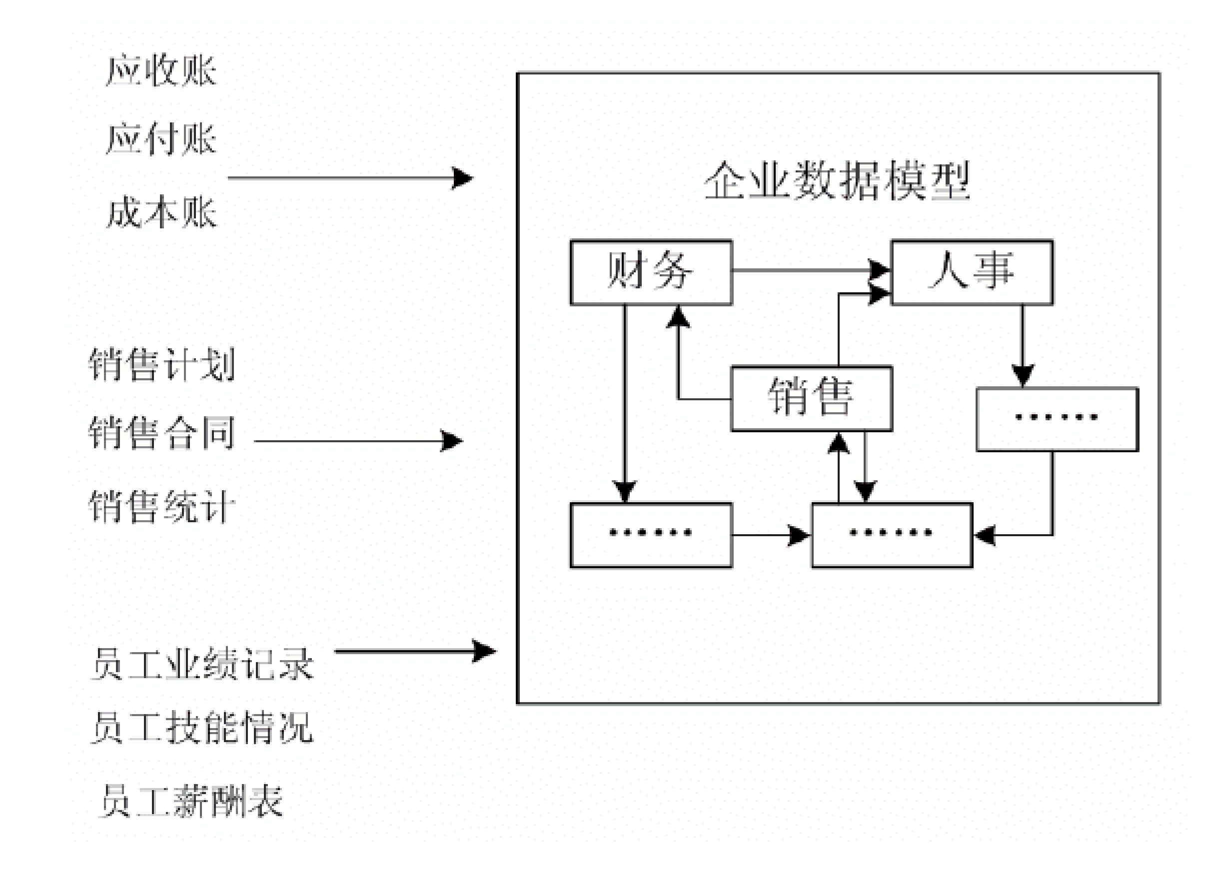


图 10.8　企业数据模型（ER 图）

表 10.3　实体与功能关系 CRUD 矩阵

	用　户	订　单	产　品	销售代表	供 货 商
订单输入	CRUD	CRUD	R	RU	RU
订单处理		CRUD		CRUD	
产品管理	R	R	RU		R
预算系统	R	R	R	RU	R
财务计算	RU	R	RU	R	R
制造控制	R	RU	CRUD		R
后勤	R	RU	R		RU
生产控制		RU			

数据仓库分析人员在数据仓库的概念模型定义中还要了解现行业务处理系统的数据存储方式，从中找到数据仓库的数据映射源的物理状况。因此，需要用数据存储模式表将所有的数据源存储模式列出。根据此表，数据仓库分析人员还需要对每个数据源进行分析：这些数据源存储模式的管理者是否为数据仓库的建设提供某种程度的支持？客户/服务器之间的连接通过哪种通信协议给予支持？数据源的存储模型使用哪些操作语言？在了解这些情况后，数据仓库设计人员可将数据仓库与特定的业务处理系统中的数据源成功地连接在一起。表 10.4 即为现行业务处理系统的数据存储模式表。

表 10.4　现行业务处理系统的数据存储模式表

	Oracle	Sysbase	SQL Server	VEP	其他存储模式
订单输入	√			√	
订单处理	√			√	
产品管理		√			
预算系统					√（Excel）
财务计算			√		
制造控制			√		
后勤				√	
生产控制			√		
外部数据源					
销售代理商				√	
市场调查公司			√		

3. 概念模型的分析

完成数据仓库概念模式的定义后，还要进一步考察模式的用户要求和系统环境，分析数据仓库范围内的主要对象，确定系统的主要主题域以及主要主题域之间的联系。数据仓库设计者通过对用户的访问，得到用户对数据仓库结构以及数据仓库存在环境的要求，并将分析结果转变成概念模式，提交给被访问者进行确认，以保证设计者对当前环境的正确性理解。

概念模型一般用 ER 图表示，图中各个对象（实体）间存在着相互的联系，用长方体表示实

体，对应于数据仓库中主题；椭圆表示主题的属性；用菱形表示主题间的联系。图 10.9 为超市数据仓库的概念模型。

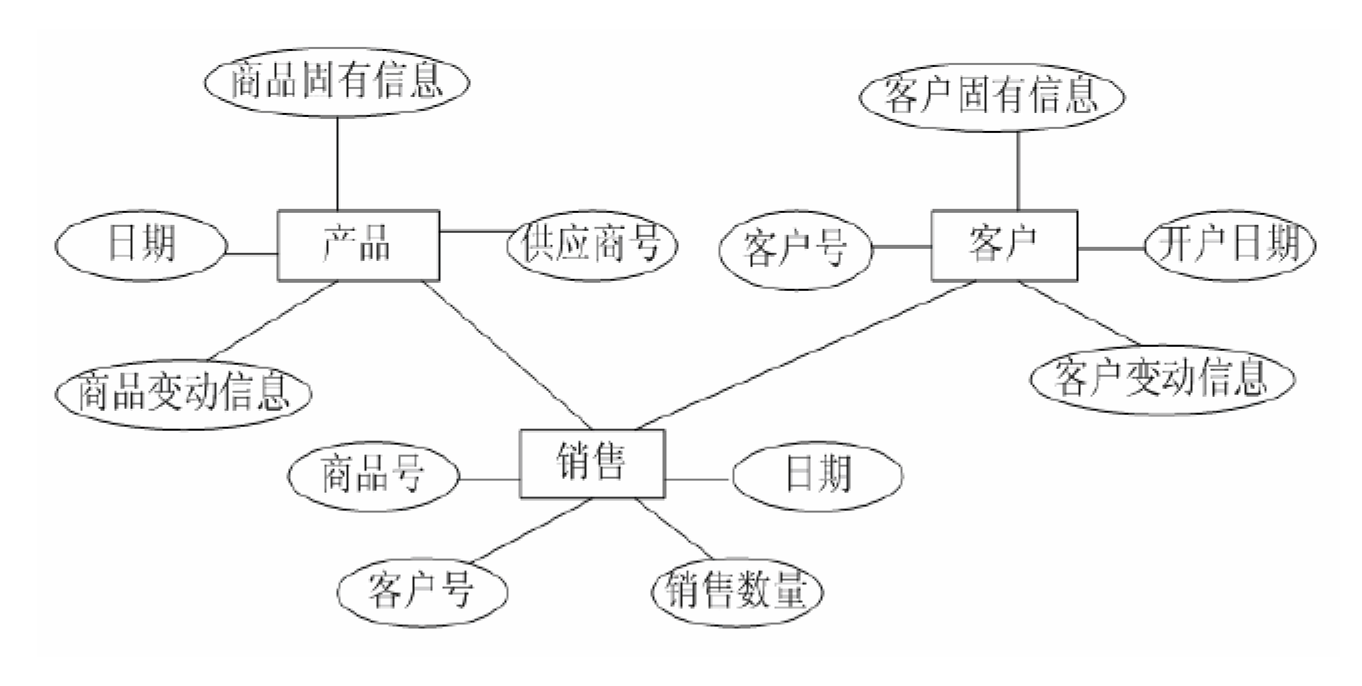
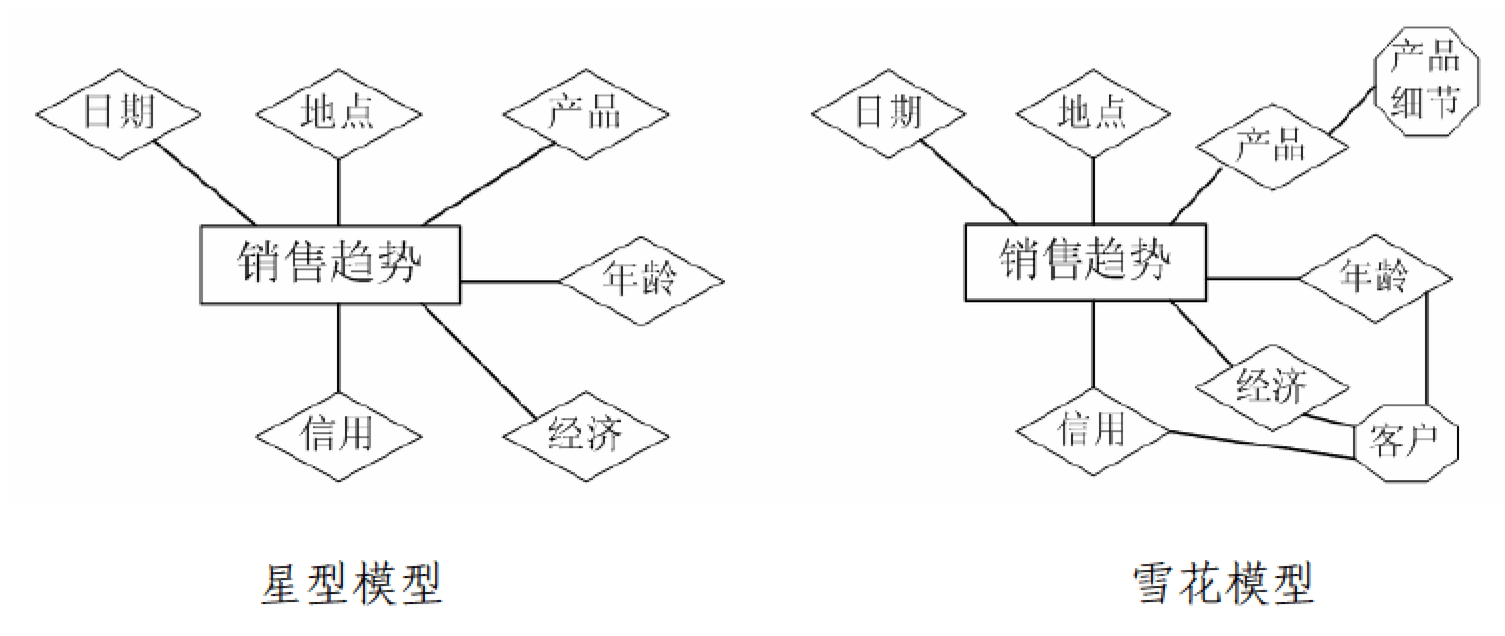


图 10.9 商品、销售和客户的概念模型

4. 概念模型的设计

图 10.9 所示的概念模型并不适合数据仓库的设计，在数据仓库的概念模型设计中，常用星型模式和雪花模型。图 10.10 即为销售主题的星型和雪花模型。



星型模型　　　　雪花模型

图 10.10 销售主题的星型、雪花模型

完成概念模型设计以后，必须编制数据仓库开发的概念模型文档，且对概念模型进行评价。文档包括数据仓库开发概念模型需求分析报告、概念模型分析报告、概念模型和概念模型的评审报告。概念模型的评审就是确定概念模型是否完整、准确地描述了用户的决策分析环境，使得数据仓库开发人员可以找到一个比较理想的数据仓库解决方案，并且能够进一步获得用户的积极支持。

10.3.2 数据仓库的逻辑模型设计

在数据仓库设计开发中无法直接依靠概念模型实现数据仓库的物理模型，还要依靠逻辑模型作为概念模型到物理模型转换的桥梁。

在进行数据仓库的逻辑模型设计时，一般需要完成分析主题域，确定装载到数据仓库的主题，确定粒度层次划分，确定数据分割策略，关系模式的定义和记录系统定义，确定数据抽取模型等。

1. 分析主题域

数据仓库的设计方法是一个循环的过程，在进行数据仓库的设计时一般是一次先建立一个主

题或几个主题。在超市数据仓库的概念模型设计时，首先确定了它的三个基本主题域：商品、销售与客户。分析后可以认为"销售"主题既是一个超市的最基本的业务对象，又是进行决策分析的最主要领域，因而可以把"销售"主题域定义为首先建立的主题。通过"销售"主题的建立，超市经营者可对整个超市的经营情况有较全面的了解，尽快地满足超市经营者建立数据仓库的最初要求。

当数据仓库中的主题定义后，也就基本构成了逻辑模型。此时，需要在主题的逻辑关系模式中包含所有的属性以及与系统相关的行为。表 10.5 即为超市数据库三个主题的属性描述。

表 10.5 主题的详细描述

主 题 名	公共码键	属 性 组
商品	商品号	商品固有信息：商品号、商品名、类型、颜色等 商品采购信息：商品号、供应商号、供应价、供应日期、供应量等 商品库存信息：商品号、库房号、库存量、日期等
销售	销售单号	销售单固有信息：销售单号、销售地址等 销售信息：客户号、商品号、销售价、销售量、销售时间等
客户	客户号	客户固有信息：客户号、客户名、性别、年龄、文化程度、住址、电话等 客户经济信息：客户号、月收入、家庭总收入等

2. 粒度层次的划分

在数据仓库的逻辑设计中还要解决的一个重要问题是决定数据仓库粒度的层次划分，粒度层次的划分适当与否直接影响到数据仓库中要存储的数据量和查询方法。

划分数据粒度，先要估算数据仓库所需要占用的存储空间，然后再依据存储空间确定粒度划分。估算数据仓库中的存储空间，可以从需要建立的表数目出发，通过估算每个表的大致行数（一般为最多和最少的行数）及每行占用空间的最大字节和最小字节数，得到表的存储空间。因为数据仓库的数据存取是通过存取索引来实现的，而索引是对应表中的行来组织的，因此通过表的行数就可得到数据仓库的数据存取索引的大小即索引空间。表的存储空间与相应的索引空间之和就为数据仓库所需要占用的存储空间。得到数据仓库的存储空间就可按表 10.6 所示进行数据粒度的划分。

表 10.6 数据仓库的存储空间与数据粒度划分策略对照表

一年数据		五年数据	
数据量（行数）	粒度划分策略	数据量（行数）	粒度划分策略
10 000 000	双重粒度并且仔细设计	20 000 000	双重粒度并且仔细设计
1 000 000	双重粒度	10 000 000	双重粒度
100 000	仔细设计	1 000 000	仔细设计
10 000	不考虑	100 000	不考虑

在数据仓库中确定粒度时，需要考虑：要接受的分析类型、可接受的数据是最低粒度、能够存储的数据量。

计划在数据仓库中进行的分析类型将直接影响数据仓库的粒度划分。将粒度的层次定义越高，就越不能在该仓库中进行更细致的分析。例如，当粒度层次定义为月份时，就不可能利用数据仓库进行按日汇总的信息分析。

数据仓库通常在同一模式中使用多重粒度。数据仓库中，可以有今年创建的数据粒度和以前创建的数据粒度。例如可用低粒度数据保存近期的财务数据和汇总数据，对时间较远的财务数据只保留粒度较大的汇总数据，这样既可以对财务近况进行细节分析，又可以利用汇总数据对财务趋势进行分析，这里的数据粒度划分策略就需要双重数据粒度。

定义数据仓库粒度的另外一个要素，是数据仓库可以使用多种存储介质的空间量，如果存储资源有一定的限制，就只能采用较高粒度的数据粒度划分策略。

选择一个合适的粒度是数据仓库设计过程中所需解决的一个复杂决定，因为粒度的确定实质上是业务决策分析、硬件、软件和数据仓库使用方法的一个折中。在确定数据仓库粒度时，可以采用多种方法做到既能满足用户决策分析的需要，又能减少数据仓库的数据量。如果主题分析的时间范围较小，可以保持最小的数据粒度，但是只保持较小时间的细节数据。例如在分析销售趋势主题中，分析人员只利用回溯一年的数据进行比较，那保存销售主题的数据只需要 15 个月的数据就足够解决问题了，而不必保存大量的、时间过长的数据。

3. 确定数据分割策略

数据的分割是指把逻辑上整体的数据分割成较小的、可以独立管理的物理单元进行存储的方法。使用数据分割便于数据的重构、重组和恢复，以提高创建索引和顺序扫描的效率。

对于超市数据仓库而言，可以按时间对数据进行分割，即将在同一时间内的数据组织在一起。如由于超市的管理者经常关心的问题是商品在某个季节的销售情况，从而将超市的销售数据按季节进行分割，可以大大减少数据检索的范围，减少物理 I/O 次数，提高系统的性能。

在确定数据分割策略时一般要考虑以下几个方面的因素。

- 数据量。数据量的大小是决定是否进行数据分割和如何分割的主要因素，如果数据量较小，可以不进行数据分割，或只用单一标准对数据进行分割。
- 数据分析处理的对象。数据分割与数据处理的对象是紧密联系的，不同主题内数据分割的标准不同。如“商品”主题内对于数据的分类更多地采用商品大类、商品小类和时间标准，而在“供应商”主题内数据分割的标准则更多地用地理位置和时间进行分割。
- 粒度分割的策略。进行数据分割设计时，更重要的是将数据分割标准与粒度层次的划分策略统一起来。例如对“商品”主题销售数据可以按时间和商品类别的组合标准进行分割。

4. 关系模型定义

数据仓库的概念模型的物理实现必然是以各种表来完成的，这些表可由指标实体转换、维实体、详细类别实体来完成。例如对图 10.11 所示的金融企业客户主题逻辑模型可以设计出不同的事实表和维度表。

客户的事实表模型有：客户事实表（基本情况表、变动情况表）、客户贷款事实表（房屋贷款情况表、汽车贷款情况表）、客户存款事实表（客户存款表 1、客户存款表 2、……）、客户担保事实表（客户担保事实表 1、客户担保事实表 2、……）。

客户维度表模型有：时间维度表（年、月、日）、地点维度表（省、市、县、街道）、贷款维度表（抵押贷款、非抵押贷款）等。

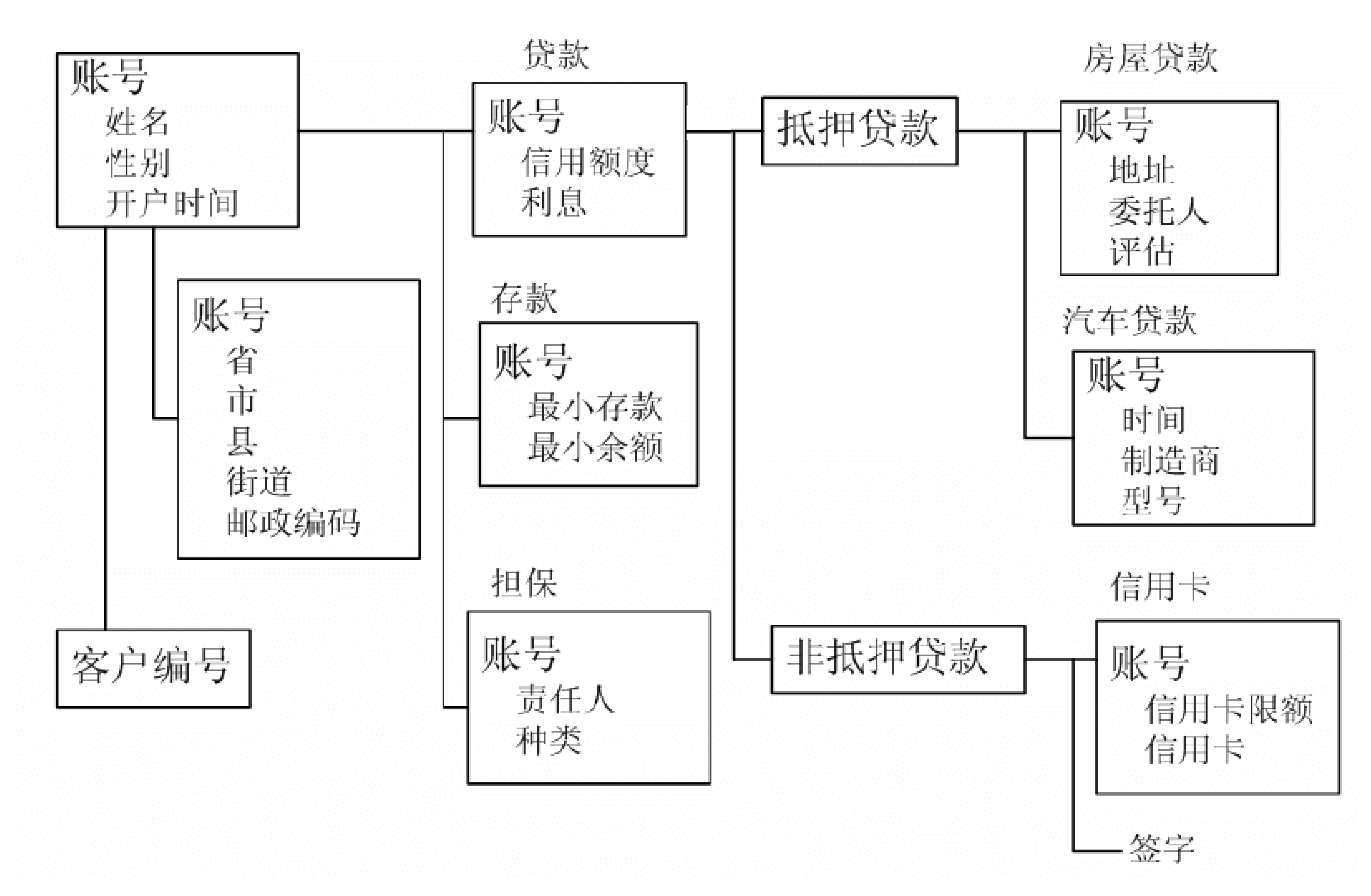


图 10.11 金融企业客户主题逻辑模型

事实表中一般包含由主键和外键所组成的键部分及与用户希望在数据仓库中所了解的数值指标；维度表则含有商业项目的文字描述，维度的设计提供了维度属性的定义。在观察维度表中的维度对象时，其属性可以看作描述该项目的各种信息。如对产品信息的维度可以用各种颜色描述。维属性在分析数据仓库中的数据时非常有用，从某种角度讲，维属性就是用户获取数据的窗口。

5. 数据仓库的实体定义

在设计逻辑模型时，不仅必须对逻辑模型中的每个实体进行具体的定义，而且要进一步确定实体列中的主键列（也即查询词），实体之间关系的外部键列，实体物理存储的一些特性。

主键列是用于识别实体实例的唯一识别数据表行的列，通常由一个列或多个列组成，有时还需要确定一些候选键列。如可以将 CustomerNumber 作为客户实体的主键列，以 CustomerName 作为候选键列，以更好地识别每个客户实体。

为在数据仓库的物理模型中表示实体之间的联系，必须确定实体的外部键列，它是存在于某个实体中的某一列或某一组列，它们的值在其他实体中作为主键处理。例如在订单细节实体中客户编号列（CustomerNumber）用于描述签订订单的客户，而 CustomerNumber 则是客户实体中的主键，此时 CustomerNumber 就成为订单细节实体的外部键列，用此键列可以将其与客户实体关联起来。

6. 数据抽取模型

数据仓库的抽取模型由数据抽取处理过程、数据源表、数据源抽取过滤条件与连接表、数据抽取过程的排序与聚集表、数据抽取的目标与源列对应关系表等组成。

数据仓库的抽取处理是传统的数据处理过程，其输入是数据仓库数据源的各种业务操作处理系统的数据库，输出部分是数据仓库。

为实现数据仓库的正确数据抽取还要利用数据抽取规则确定从哪些数据源中抽取哪些数据，这些数据基于什么样的数据平台，即数据源抽取对象。

在数据的抽取分析中还需要分析所抽取的数据应该满足哪些条件，这些条件可能是一些复合条件，而且可能来自不同的表。表 10.7 即为数据源抽取规则表。

表 10.7 数据源抽取规则表

表 列 名	过滤与连接条件	比 较 值	复合条件	备 注
aa1	<	50000	AND	采购商品数量小于 50000
aa1	>	500	AND	采购商品数量大于 500
aa1	≠	'AB'	OR	商品前两位非'AB'
……	……	……	……	……

将数据从数据源抽取到数据准备区后，还需要对所抽取的数据进行各种清理工作，这些数据的清理内容必须在逻辑模型设计过程中确定下来。数据的清理内容可以包含数据类型的转换，例如将整型数据转变为实数类型，或将数据的日历格式进行统一，或将数据值中按照粒度模型进行汇总、聚集处理，如可以对数据进行按一定的规则排序及分组。

当完成数据的排序与分组之后，就可以将数据从数据准备区加载到数据仓库中，即将数据源加载到数据仓库中的相应目标数据列上。

在完成数据仓库的逻辑设计后，应该将逻辑模型设计方案整理成文档，并且组织有关人员对其进行评审。评审主要集中在主题域是否可以正确地反映用户的决策分析需求。

10.3.3 数据仓库物理模型的设计

数据仓库的物理模型是逻辑模型在数据仓库中的实现模式，其中包括逻辑模型中各种实体表的具体化，例如表的数据结构类型、索引策略、数据存入位置以及数据存储分配等。

1. 数据仓库设计的规范

由于在数据仓库中包括多种表、列与域等，为保证数据仓库的设计、实施和管理保持稳定，不产生混乱，需要对物理数据模型中的实体、表、列等进行规范化处理。规范化的内容主要有完整清晰的数据定义、合适的数据格式等。

完整清晰的数据定义能使数据仓库开发人员和用户很清晰地了解所定义的数据，在尽可能的情况下采用完整的定义，或者使用一些常用的缩写方式，例如客户编号可以使用 CustomerNumber 或 CusNo。对于数据定义的格式必须大小一致，为提高数据定义的可读性可以采用大小写混合方式，在使用比较长的字符描述数据定义时，可以采用适当的下画线或连字符来提高数据定义的可读性。

2. 确定数据结构的类型

在数据仓库的结构中，可能包含各种数据类型的任意组合：细节数据、概括数据、外部数据、多维数据、数据子集、专门数据缓存、复制数据和存档数据。数据仓库的设计人员必须确定符合设计目标的数据结构类型。图 10.12 列出了各种数据类型及其关系。

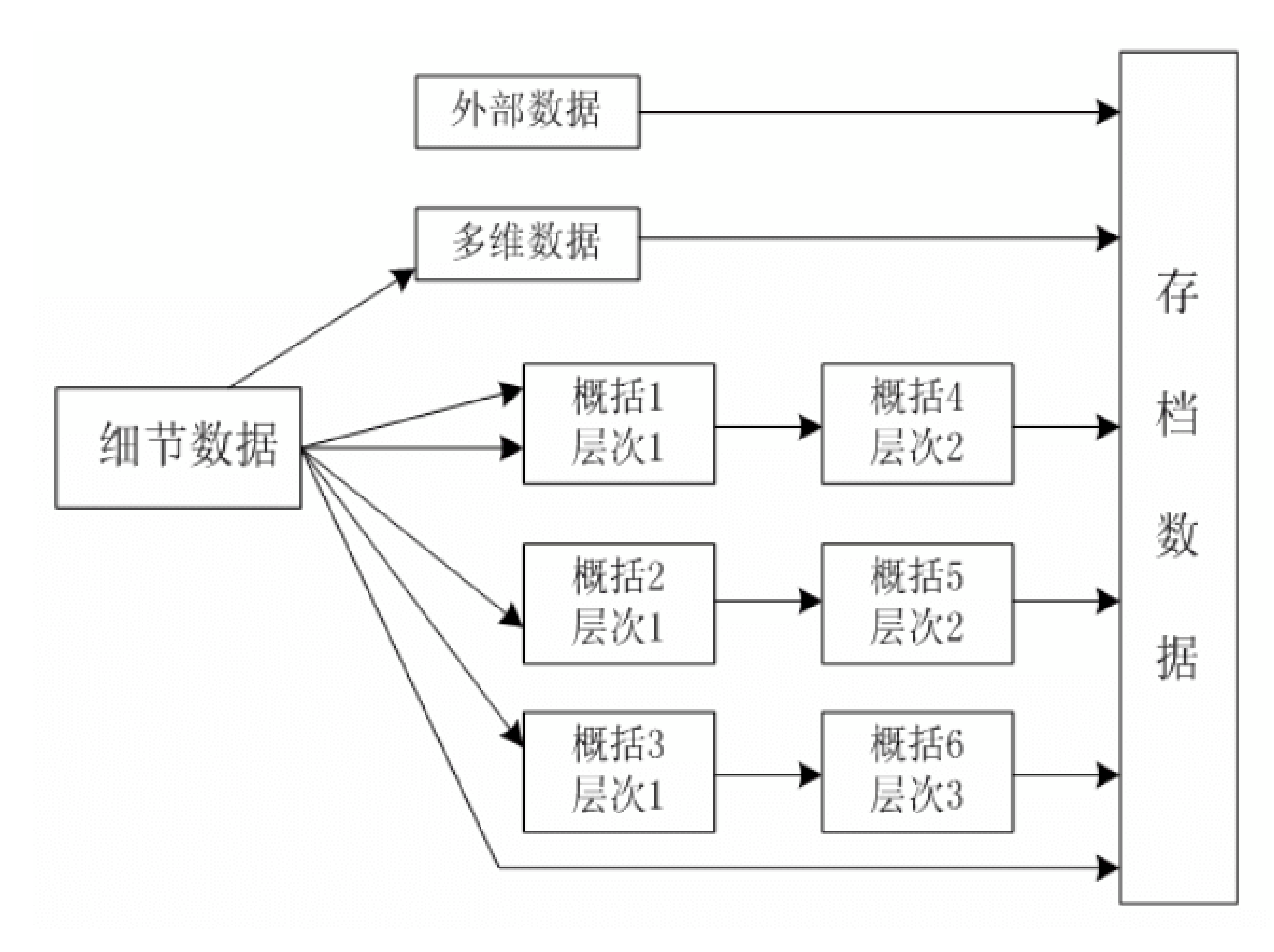


图 10.12　数据仓库的数据结构类型

虽然数据仓库的基础是规范化的数据模型，但在数据仓库中为了提高数据仓库的运行效率，需要进行数据的非正规化处理。例如可以将“最后订货日期”和“最后发货日期”等字段加入数据仓库，可以提高查询效率。

3. 确定索引策略

数据仓库的数据量很大，要对数据的存取路径进行仔细的设计和选择。由于数据仓库的数据一般很少更新，因而可以设计索引结构来提高数据存取效率。在数据仓库中，设计人员可以考虑对各个存储建立专用的、复杂的索引，以获取较高的存取效率。一般都按主关键词或大多数外部关键词建立索引，可以按照索引使用的频率，由高到低逐步添加，直至某个索引加入后，使数据加载或重组表的时间过长时，就结束索引的添加。

4. 确定数据存放位置

同一个主题的数据并不要求存放在相同的介质上，在物理设计时，常常根据数据的重要性、使用频率以及对响应时间的要求进行分类，且将不同类型的数据分别存储在不同的存储设备中。重要性高、经常存取、对响应时间要求高的数据存放在高速存储设备上。

在设计数据的布局时要注意遵循以下 5 个原则。

- 不要把经常需要连接的几张表放在同一设备上。
- 把要进行公共连接的表放在同一服务器上。
- 如果几台服务器之间的连接会造成严重的网络业务量的问题，则要考虑服务器复制表格。
- 考虑把整个企业共享的细节数据放在主机或其他集中式服务器上。
- 别把表格和它们的索引放在同一设备上。

5. 确定存储分配

在数据仓库的物理模型设计中，需要确定不同数据的存储分配。数据可集中在一台服务器上，也可以按工作小组部门、主题区或应用程序分散在多个服务器上。按照部门或工作小组进行数据

分区时，各个部门数据的数据结构是针对每个部门具体的用户群而定的。

在完成物理模型设计后，要对设计的物理模型进行评审，它主要涉及所有的数据定义语言、联机过程或批过程的描述，已知的预期数据使用情况，数据量和事务量，预计的数据增长速度及物理设计文档，以获得物理模型在满足数据仓库使用的灵活性，性能、数据完整性，系统可用性、数据的当前性和用户的满意度等方面的结果。

10.4 数据仓库的技术管理

数据仓库在创建后，通过测试就可以进行使用阶段，在使用阶段中需要不断加强对数据仓库的技术管理。这些技术管理工作涉及以下几方面。

1. 数据加载的一些问题

（1）数据准备区。由于数据仓库的数据抽取、清理、加载需要较长的工作时间，因此常常设置一个数据准备区的临时数据库，以用于这些操作。在数据准备区中可以设置数据抽取、清理和加载的重新启动机制，以避免这些操作失败后可以从失败处重新启动而不必从头开始。为此可以将数据的抽取、清理和加载活动分成若干步骤，且在进入某个步骤后，保留当前的状况。

（2）数据加载方式的选择。数据加载的方式一般批处理，而且数据的加载一般选择在节假日或夜间进行。

（3）大批量数据加载的处理。大量数据加载往往导致数据的刷新，这对数据仓库而言是不容许的。因此，大量数据的加载与刷新活动只能在数据仓库刚建立后的第一次数据加载的活动中进行，以后的数据加载往往需要采用增量数据加载方法。大批量数据加载可以采用数据复制技术实现，它可以保证数据加载过程中的完整性约束，不会受到系统失败等不良因素的影响，并且对数据的传送进行优化处理。

2. 故障恢复管理

数据仓库一旦开始运行，来自管理方面和用户方面不断进行存取的压力也会增加，因此需要制订故障恢复规划。在故障恢复规划中可以采用的步骤如下：

（1）停止包括操作系统在内的服务器；

（2）重新安装和重新配置操作系统；

（3）重新标定驱动器；

（4）重新安装和重新配置关系数据库系统、监控程序和中间件；

（5）对数据重新加载和重新索引。

3. 访问控制与安全管理

为了保证数据仓库中数据的安全，需要控制对数据仓库的访问。可以采用多种方法实现数据仓库的安全性。如对细剖能力（即从高度概括的数据入手，不断访问详细的数据）进行限制，且对特定的概括数据表和运行的详细内容提供访问控制，并且还要限制对数据源的使用，如创建临时表和即席表查询等。当一个用户离开时，净化程序必须消除对多个系统的访问控制。

4. 数据增长的管理

数据仓库存储的数据量非常大，远远大于运行数据库的数据量，因此需要利用一些通用的商业和管理实践，控制和管理数据量的增加。

（1）概括技术。对细化数据进行高度概括可以明显地减少数据量，但为了提供细剖数据的能力，需要将细化数据存储起来。

（2）对细剖数据的控制。控制细剖的程度可以大大减少数据量。

（3）历史数据的限制。限制必须存储到数据仓库中的历史数据的长度，只选择在现阶段仍然有效或有借鉴意义的那些历史数据。

（4）数据使用范围的限制。利用能够改变收集数据环境的商业事件知识限制管理的数据范围。例如当两个公司合并时，它们各自的历史数据的价值可以是不同的。

（5）睡眠数据的移出。在数据仓库的使用过程中，可能会产生大量的睡眠数据，例如对决策没有价值的数值、超出特定时间的历史数据等。随着睡眠数据的增加，可用于查询处理的实际可用数据百分比在不断降低，最后导致数据仓库的使用效率急剧下降。

解决这个问题的一种办法就是找出并移出查询时很少用到的数据；或采用邻线存储系统的二级存储模式。邻线存储系统就是一种处于在线和离线之间的存储系统，这种系统虽然不是在线联机状态，但是可以为用户提供一个合理的访问时间。

10.5　OLAP 技术

数据仓库是一种管理决策分析的基础，若要有效地利用数据仓库的信息资源，须有强大的工具对数据仓库中的信息进行分析决策。在线分析处理或联机分析处理（On-Line Analytical Processing，OLAP）就是一个得到广泛应用的数据仓库使用技术。

OLAP 专门用于支持复杂的决策分析，支持信息管理和业务管理人员决策活动的一种决策分析工具，它可以根据分析人员的要求，迅速、灵活地对大量数据进行复杂的查询处理，并且可以直观的、容易理解的形式将查询结果提供给各决策人员，以便能迅速、准确地掌握企业的运营情况，了解市场的需求。

OLAP 技术主要有两个特点：一是在线性，表现为对用户请求的快速响应和交互式操作；一是多维分析，能够提供对数据分析的多维视图和分析，包括对层次维和多重层次维的支持。

10.5.1　基本概念

在 OLAP 中有维、维的层次、维成员、多维数据集、数据单元、多维数据集的度量值等概念，其中维和维的层次概念在前面已有介绍，在此主要介绍其他概念。

1. 维成员

维成员是维的一个取值，如果维分成了若干个维，那维成员就是不同维层次取值的组合。如“陕西省西安市新城区”就构成了地理维的一个维成员。维成员并不一定要在维的每一个层次上都取值。实际上维成员的值并不是数据仓库中所关心的对象，一般是用此值去描述真正关心的对

象即主题在维的对象。例如企业的销售人员只对销售数据感兴趣，但是在观察销售数据时，却需要以地理位置维、时间维或产品维的维成员去描述销售数据。

2. 多维数据集

多维数据集是决策支持的支柱，也是 OLAP 的核心，有时也称立方体或超立方。OLAP 展现在用户面前的是一幅幅多维视图。多维数据集可以用一个多维数组或多维表表示，对于二、三维数据集则可用相应的可视化方式表示。

3. 数据单元

多维数据集的取值为数据单元。当在多维数据集中的每个维都选中一个维成员后，这些维成员的组合就唯一确定了观察变量的值，即可表示为

（维 1 维成员,维 2 维成员,……,维 n 维成员）

4. 多维数据集的度量值

在多维数据集中有一组度量值（数值），这些值是基于多维数据集中事实表的一列或多列，一般是销售量、成本和费用等。

10.5.2 多维分析

OLAP 的多维分析是指对多维数据集中的数据用切片、切块、旋转等方式分析数据，使用户从多个角度、多个侧面去观察数据仓库中的数据，这样才能深入地了解数据仓库中数据所蕴含的信息。

1. 多维的切片

对多维数据集中的某个维选定一维成员的选择操作可以称为切片。切片数量的多少是由所选定的那个维的维成员的多寡所决定的，通过切片可以更好地了解多维数据集，并降低多维数据集的维度。

2. 多维的切块

与切片类似，如果在一个多维数据集上对两个及以上的维选定维成员的操作称为切块。很明显，切块操作可以看成进行多次切片操作以后，将每次切片操作所得到的切片重叠在一起而形成。

3. 旋转

多维数据集的旋转操作就是在对数据仓库中的多维数据集改变其显示的维方向。这种旋转操作可将多维数据集中的不同维进行交换显示，使之更加直观地显示不同维之间的关系。

4. 其他 OLAP 操作

在 OLAP 的分析中，对多个事实表进行查询，即为“钻过”操作；而在对立方体操作时，利

用数据库关系，钻透立方体的底层，进入后层的关系表的操作即为“钻透”；通过一个维的概念分层向上攀升或者通过维归纳，在数据立方体上进行聚集即为“上卷”；通过沿维的概念分层向下或引入新的维获得由不太详细的数据到更详细的数据即为“下钻”操作。

在 OLAP 的其他操作还有统计表中最高值和最低值的项数，计算平均值、增长率、利润、投资汇报率等统计计算。OLAP 还提供了分析建模机制，包括推导比率、变差等以及跨越多维计算度量的计算引擎，它能在每一粒度级和在所有维的交叉产生汇总、聚集和分层，也支持预报、趋势分析和统计分析的函数模型。

10.5.3　维的层次关系

维的层次关系可用一个层次图表示，图 10.13 即为销售地区维的层次关系图。

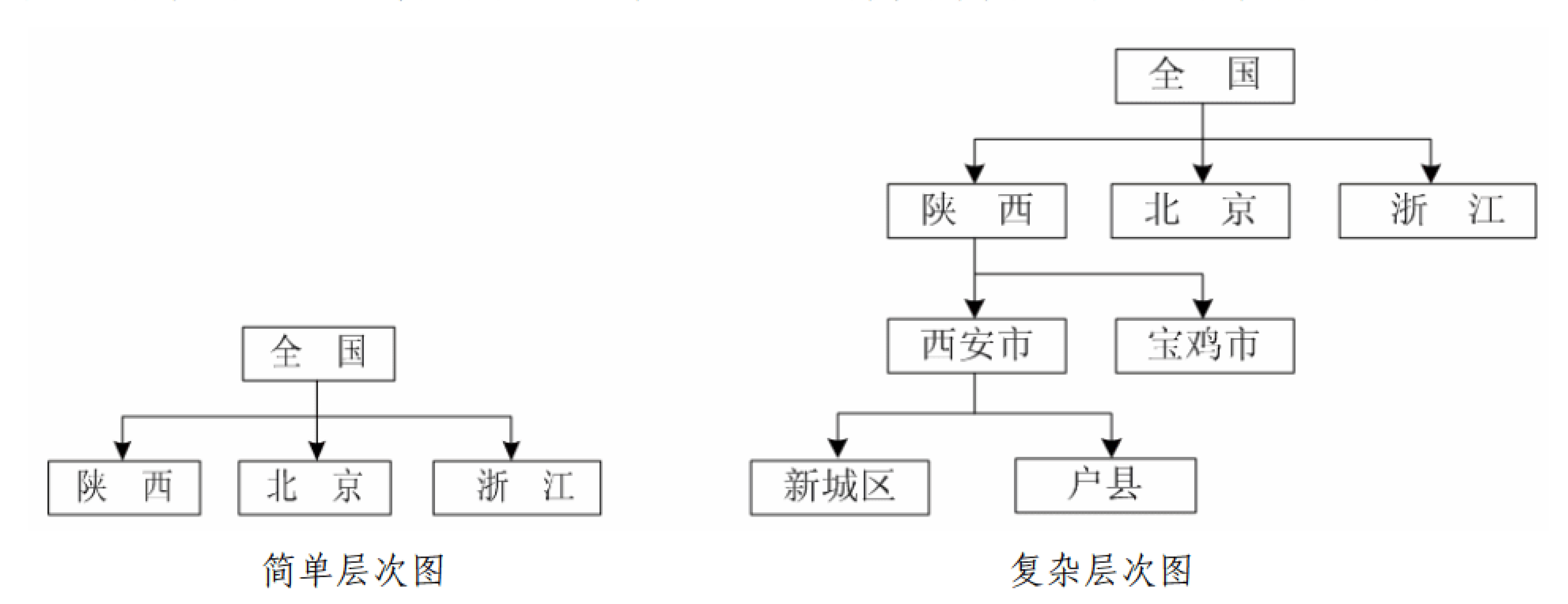


图 10.13　销售地区维的层次关系图

10.5.4　维的类关系

在 OLAP 的应用中，常常涉及对维成员的分类与归纳，即在查询中根据用户关于类别的要求对所有维成员进行分类，在分类的基础上归纳出类的共同特征或区别于其他类的特征。

在 OLAP 的应用中，有的需要按照维的层次关系进行分析，有的需要按照维成员的类进行分析。维层次分析主要从高层维到低层维的“钻取”分析和由低层维到高层维的“汇总”分析；维成员的分类归纳是指对同一层次的维成员进行聚类分析。在实际应用中，这两种方法常交叉使用，图 10.14 即为这二者的组合图。

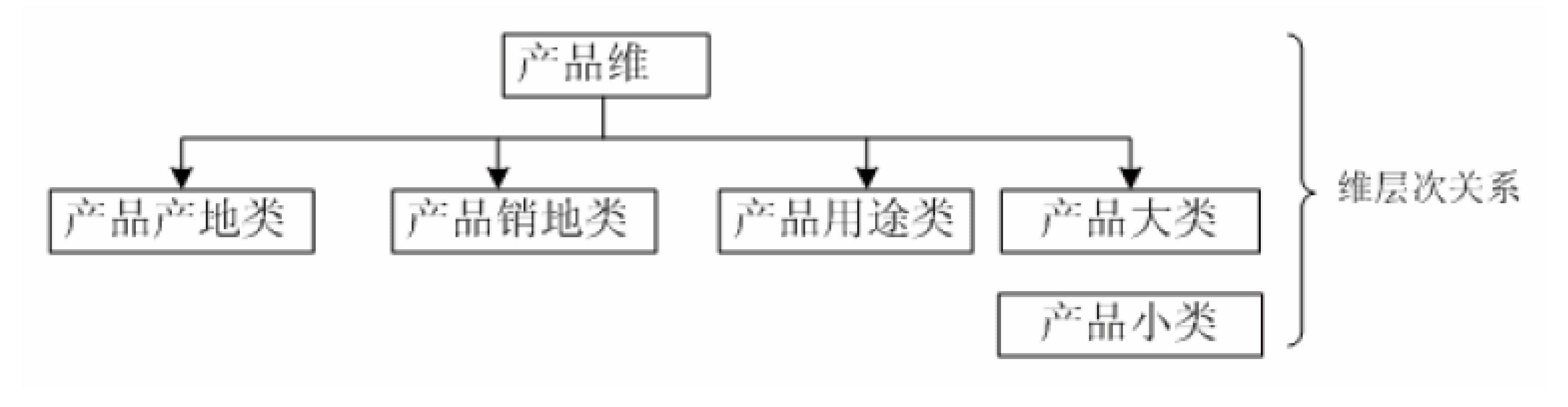


图 10.14　维的层次与类组合图

10.5.5　OLAP 与数据仓库的关系

在数据仓库中，OLAP 与数据仓库是密不可分的，但是两者具有不同的概念。OLAP 属于数据仓库应用，它以数据仓库为基础，它采用客户机/服务器体系结构，图 10.15 即为两者的关系图。

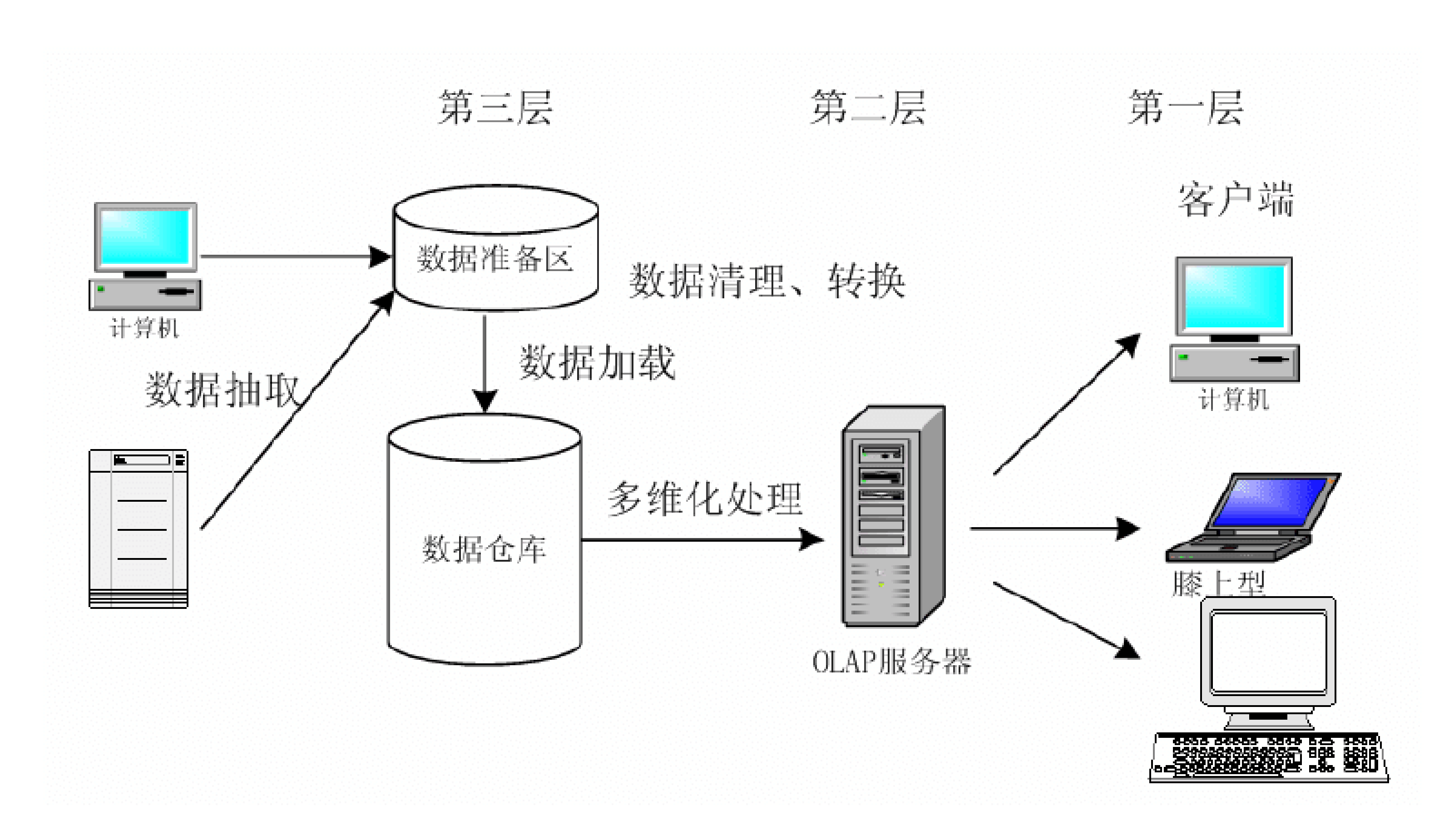


图 10.15　数据仓库与 OLAP 关系图

OLAP 采用客户机/服务器体系结构，如图 10.16 所示。它分为三层，其中第一层为客户机，实现最终用户功能，能够方便地浏览数据仓库中的数据，能够生成数据立方体，支持各种 OLAP 操作，实施决策；第二层为分析服务器，存储数据仓库中的综合数据，形成多维分析模型；第三层为企业服务器，存储数据仓库中的细节数据，它来自数据库。

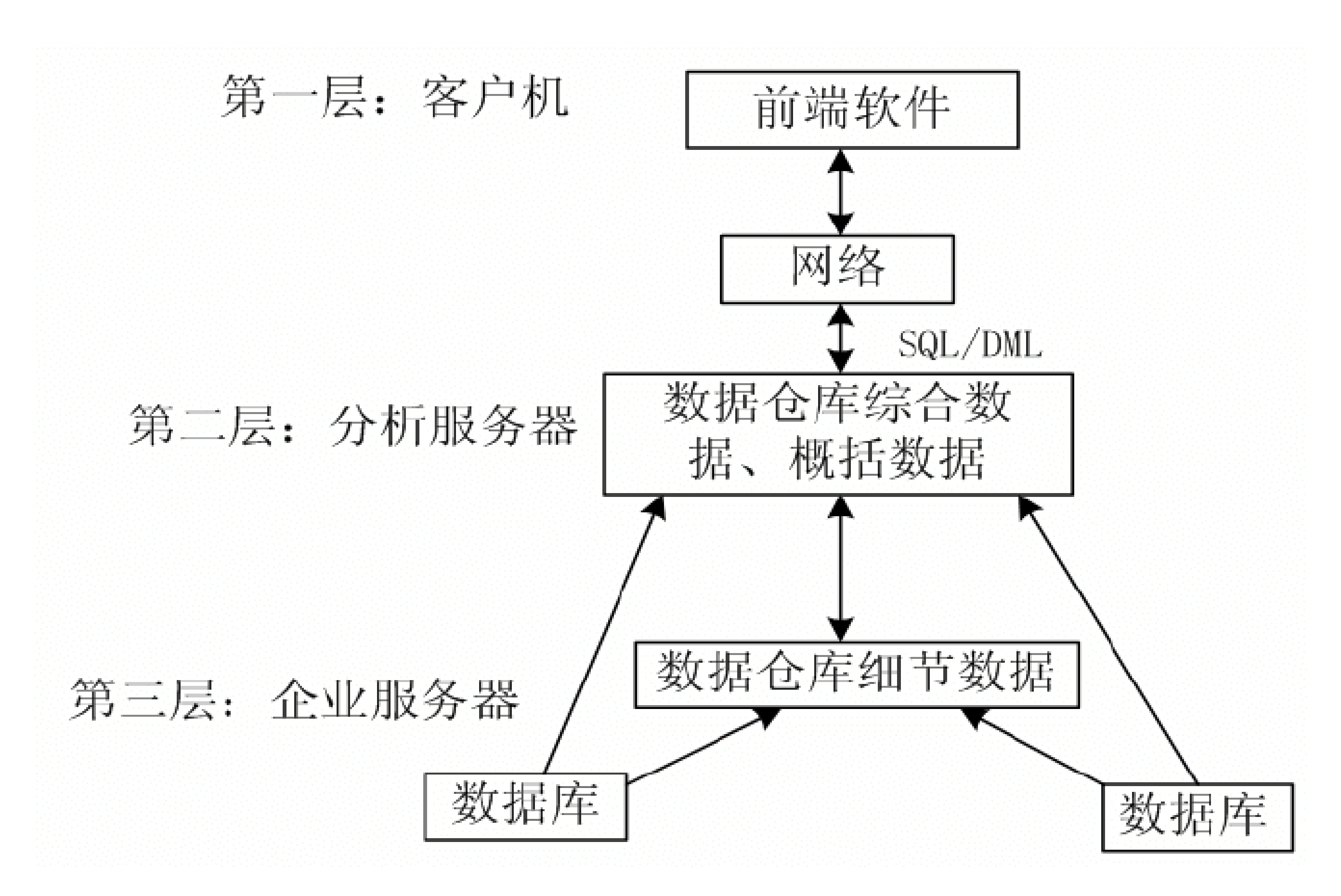


图 10.16　体系结构示意图

数据仓库的结构直接影响立方体的设计和构造，也影响 OLAP 的工作效率。为了提高 OLAP 使用的效率，在设计数据仓库时要注意以下几点。

（1）尽可能使用星型构架，如果采用雪花结构，就要最小化事实表底层维度以后的维度数量。

（2）为用户设计包含事实表的维度表，这些维度表应该包含有意义的、用户希望了解的信息。

（3）维度表的设计应该符合通常意义上的范式约束，维度表中不能出现无关的数据。

（4）事实表中不能包含汇总数据，事实表中所包含的用户需要访问的数据应该具有必需的粒度，这些数据应该是同一层次的数据。

（5）对事实表和维度表的关键词必须创建索引，同一种数据尽可能使用一个事实表。

（6）保证数据的参考完整性，使事实表中的所有数据都出现在所有的维度表中，避免遗漏事实表中的某些数据行。

10.6　基于 MATLAB 的数据仓库开发技术

数据仓库是在传统数据库的基础上发展而来的，掌握基于 MATLAB 的数据库技术，再结合 MATLAB 计算和分析工具，就可以利用 MATLAB 建立数据仓库。

MATLAB 的数据库技术主要基于数据库工具箱（Database Toolbox），它能够使 MATLAB 与通用关系数据库进行数据交流。使用数据库工具箱，可以从一个数据库将数据读到 MATLAB 工作区，然后用 matlab 的计算和分析工具处理数据，并且把结果保存到原来的数据库或另一个数据库。

10.6.1　数据库工具箱

MATLAB 数据库工具箱包括两部分内容：数据库工具箱函数和 VQB（Visual Query Build，可视查询生成器）。

1. VQB

VQB 是一个图形用户界面，用来在数据库与 MATLAB 之间交换数据，是很容易使用的工具。如果要建立查询，从数据库取数据，只要在界面上选择适当的信息即可，不需要使用函数或做过多的输入。VQB 从数据库读取数据，并把它放在 MATLAB 的单元数组、结构数组或数字矩阵中，然后用 MATLAB 的函数处理它们，还可以利用关系表、报表或图表的形式显示它们。当然，也能用 VQB 从 MATLAB 将数据输出到数据库。

2. 数据库工具箱函数

数据库工具箱函数的功能比 VQB 强大，VQB 不能完成的某些工作，函数可以完成；而 VQB 可以完成的工作，函数同样能完成。

数据库工具箱函数的功能，包括连接/关闭连接数据库，数据库中的数据可以在 MATLAB 工作区与数据库双向流动等。

数据库工具箱可以同时打开多个数据库，可以从一个数据库输出数据到 MATLAB 工作窗口，经过 MATLAB 的快速数据分析后，然后输入到另一个数据库。数据库中的多种数据格式都可以自动保存在 MATLAB 中。

数据库工具箱支持适合于 ODBC/JDBC 数据库管理系统的数据库。这样的数据库包括：IBM DB2、Informix、Ingres、Microsoft Access、Microsoft Excel、MySQL、Microsoft SQL Server、Oracle、Postgre、Sybase SQL Server、Sybase SQL Anywhere。

3. 建立数据库连接

在应用数据库工具箱前，必须利用数据库连接（ODBC）驱动程序连接相应的数据库，即建立数据源。ODBC 驱动程序是一个标准的 PV 接口，它能够使数据库管理系统与基于 SQL 的应用程序进行通信。

在此以 MATLAB 与 Access 的示例数据库 tutorial.mdb、NorthWind 为例说明数据源建立的方法。

建立数据源，是在操作系统提供的“ODBC 数据源管理器”中进行的。在 MATLAB 工作区中，输入以下命令打开可视查询生成器：

```
>> querybuilder
```

单击图 10.17 所示的 VQB 界面中的 Query 菜单项，选择其中的 Define ODBC Data Source 命令，打开如图 10.18 所示的“ODBC 数据源管理器”对话框。

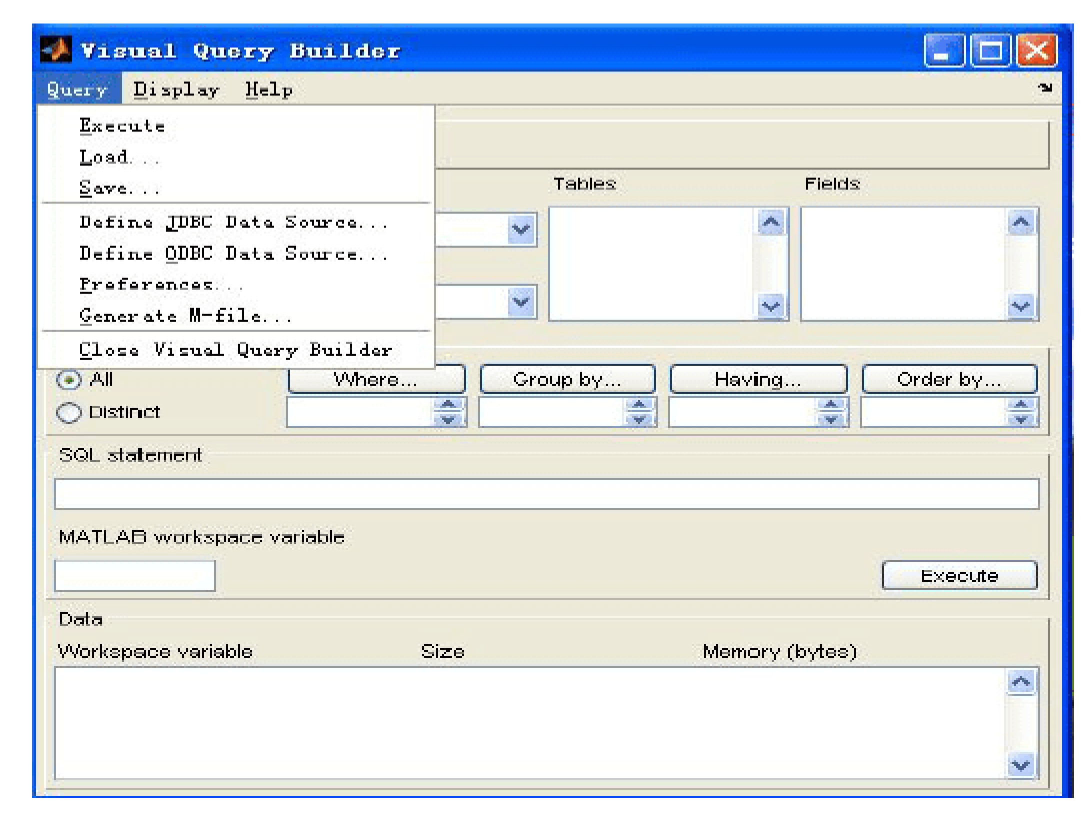


图 10.17 VQB 图形用户界面

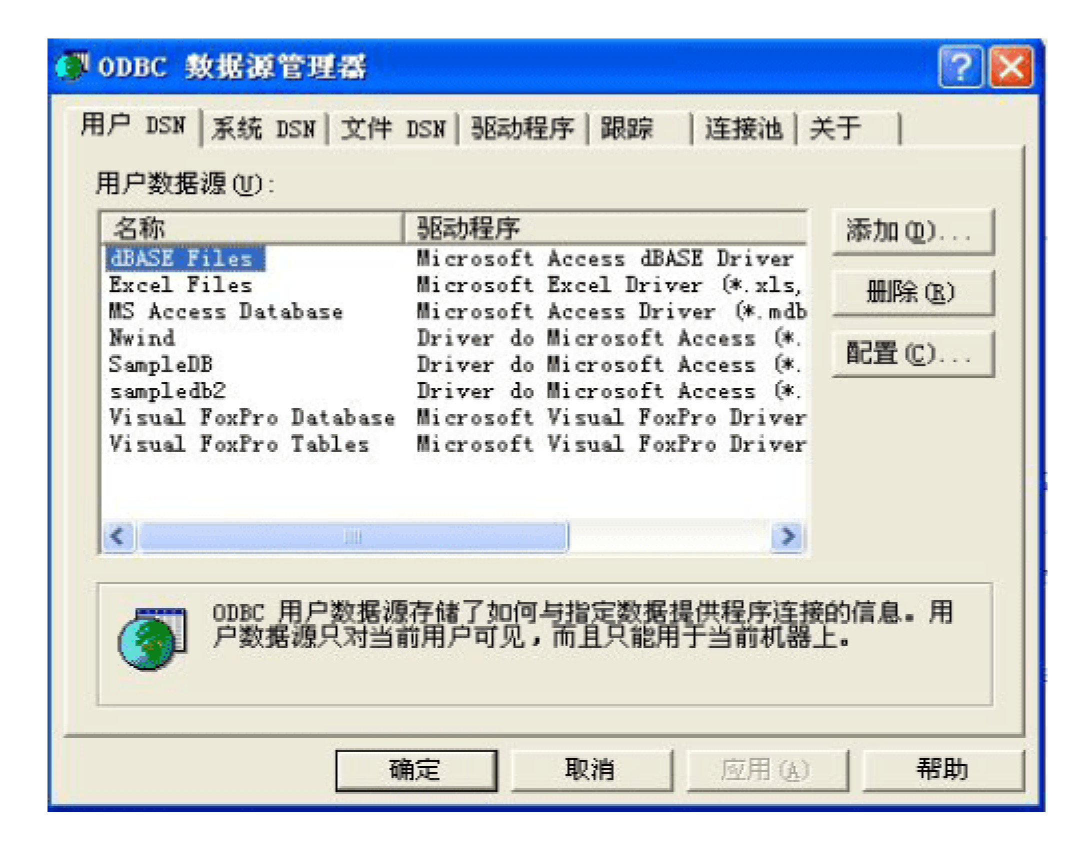


图 10.18 “ODBC 数据源管理器”对话框

在“用户 DSN”选项卡中单击“添加”按钮，打开图 10.19 所示的“创建新数据源”对话框，选择一个驱动程序（如 Microsoft Access Driver(*.mdb)），单击“完成”按钮，关闭窗口，并弹出图 10.20 所示的“ODBC Microsoft Access 安装”对话框，在其“数据源名”文本框中输入一个名字，如 sampledb，并在“数据库”选项组中单击“选择”按钮，打开图 10.21 所示的“选择数据库”对话框。



图 10.19 “创建新数据源”对话框

图 10.20 “ODBC Microsoft Access 安装”对话框

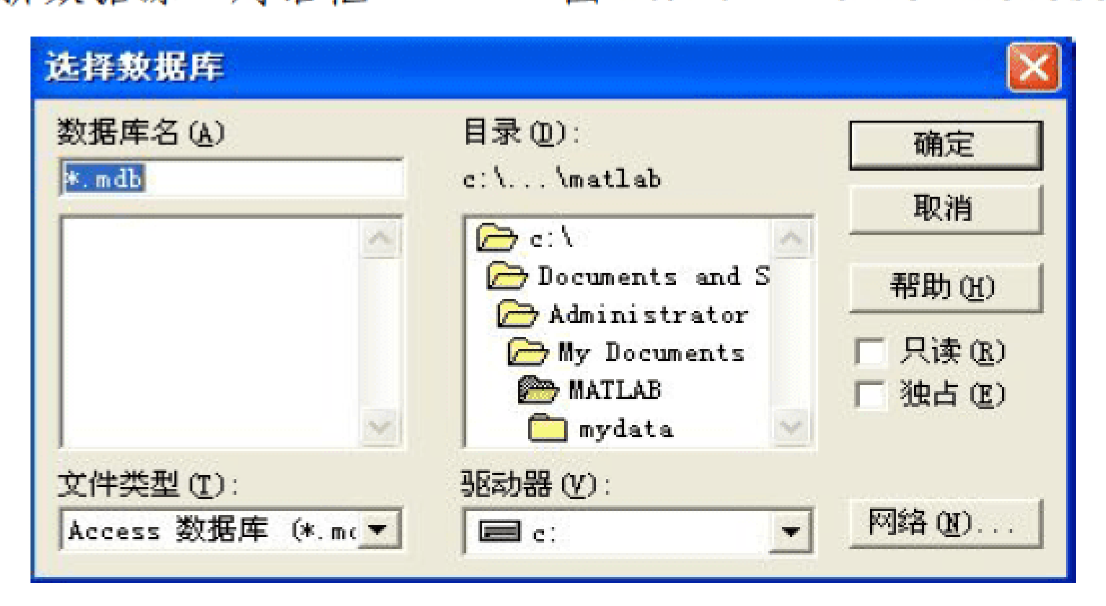


图 10.21 “选择数据库”对话框

在指定的目录中选择数据库，并单击“确定”按钮，关闭窗口，并返回“ODBC Microsoft Access 安装”对话框，单击“确定”按钮，关闭此对话框，打开“ODBC 数据源管理器”对话框，单击“确定”按钮，完成数据源的建立，此时可以在“ODBC 数据源管理器”对话框中看到刚刚添加的数据源名，只要不删除，它们就保存在数据源管理器中。

4. 数据库函数

数据库函数如表 10.8 所示。各函数的具体应用格式可参见相应的说明。

表 10.8　数据库函数

函　数	说　明	函　数	说　明
logintimeout	设置最大连接数据库的时间，单位为秒	database	连接数据库到 MATLAB
ping	得到数据库连接对象 conn 的状态信息	exec	建立数据集的游标
setdbprefs	设置数据格式，错误处理和 NULL 的优先权	fetch	输入数据到 MATLAB
close	关闭数据库连接、游标和 resultset 的对象	rows	得到输入数据的行数
cols	得到输入数据的列数	width	得到输入数据集的列宽度
columnnnames	得到数据集的列名	attr	得到输入数据的列的属性信息
get	得到对象的属性	insert	从 MATLAB 添加数据到数据库
commit	函数确认数据库的改变	updata	用 MATLAB 的数据替换数据库表中的数据
rollback	撤销对数据库的修改	set	为对象设置属性值
dmd	构造数据库 metadata 对象	supports	测试数据库元数据对象是否支持某种属性
table	得到数据库的表名	close	关闭连接

10.6.2　可视查询生成器

可视查询生成器（Visual Query Build，VQB）是一个非常容易使用的图形用户界面，用来与数据库交换数据。在 VQB 中，可以通过从列表中选择信息以建立查询，读取数据。VQB 从数据库读取数据，把它放在 MATLAB 的单元数组、结构数组或数字矩阵中，这样可以用 MATLAB 函数处理它。使用 VQB 能以关系表、图表和报表的形式显示数据，也能从 MATLAB 输出数据到数据库，生成新记录。

1. 建立数据源

与数据库工具箱函数一样，VQB 使用前，也要先建立数据源，其建立方法与此相同，在此不再赘述。

2. 启动与关闭 VQB

启动 VQB，只要在 MATLAB 工作区中输入以下命令或选择 MATLAB 窗口的 Start 菜单项即可：

```
>> querybuilder
```

关闭 VQB，使用 Query 菜单项中的 Exit 命令，或者直接关闭 VQB 图形界面。

3. 建立并执行输入数据的查询

VQB 窗口如图 10.22 所示，其中窗口各元素的意义如下。

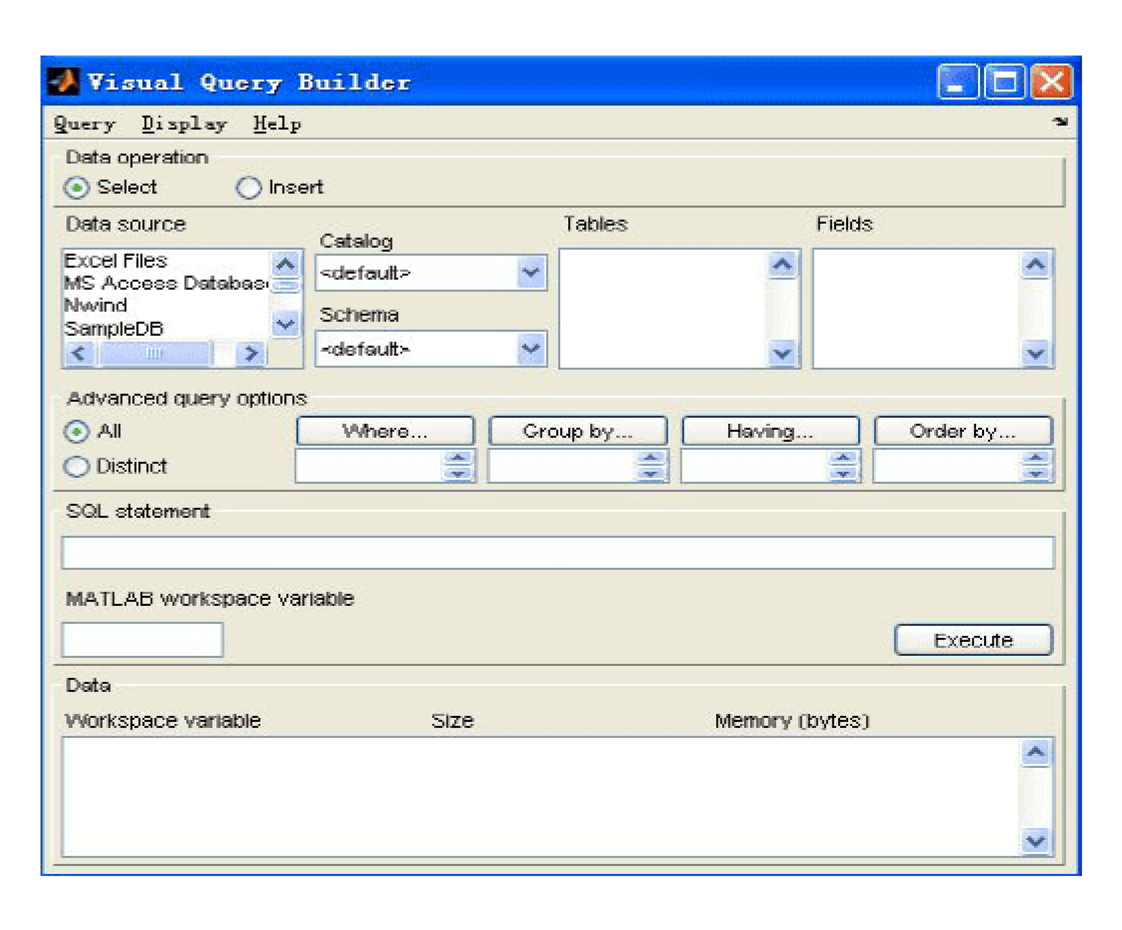


图 10.22 VQB 窗口

- Query：为读取数据设置格式优选权：保存、加载、执行查询。
- Display：显示结果，有关系表、图表、报表等形式。
- Data operation：指定选择。输入数据时选择“Select”；输出数据时选择“Insert”，为必选项。
- Data source：指定数据源，必选项。
- Table、Field：选择表和字段，必选项。
- Advanced query options：精细查询选项。
- SQL statement：自动生成的语句。
- Matlab workspace variable：为输入数据定义变量，若是数组，则输入的数据将放在这里，必选项。
- Execute：执行查询，必选项。
- Workspace variable：输入数据的概况显示：变量名、数组大小、占内存字节数。双击变量名可在数组编辑窗口中查看结果。

4. 为读取数据建立查询并执行

以下均假设已经建立了数据源 dbtoolboxdemo，赋予它的数据库是 tutorial。

打开 VQB：

```
>> querybuilder
```

然后在 VQB 窗口按下列步骤操作：

① 在 Data operation 域选择“Select”，表示从数据库选择数据。

② 从 Data source 列表框中选择数据源以输入数据。此后在 Table 列表框中将出现 tutorial 中的表名。

③ 从 Table 列表框中选择要输入数据的表 salesaVolume，此后表中的字段将出现在 Fields 列表框中。

④ 同时选择字段名：StockNumber、January、February 和 March（按住 Ctrl 键的同时，单击这个字段名），要从这些字段读取数据，此时，生成的查询语句出现在 SQL statement 域中。

⑤ 在 matlab Workspace variable 中为查询返回的数据指定一个变量名 A。

⑥ 单击 Execute 按钮，执行查询。读取的数据存在 MATLAB 的变量 A 中，A 是一个单元数组（默认值），它的信息显示在 Data 域中，如图 10.23 所示。可以用 setdbprefs 改变 A 显示的格式。

⑦ 双击 data 域中的 A，其内容显示在数组编辑器（Array Editer）中，如图 10.24 所示。同样如果在 MATLAB 工作窗口输入变量名 A 也可以显示读取的数据。

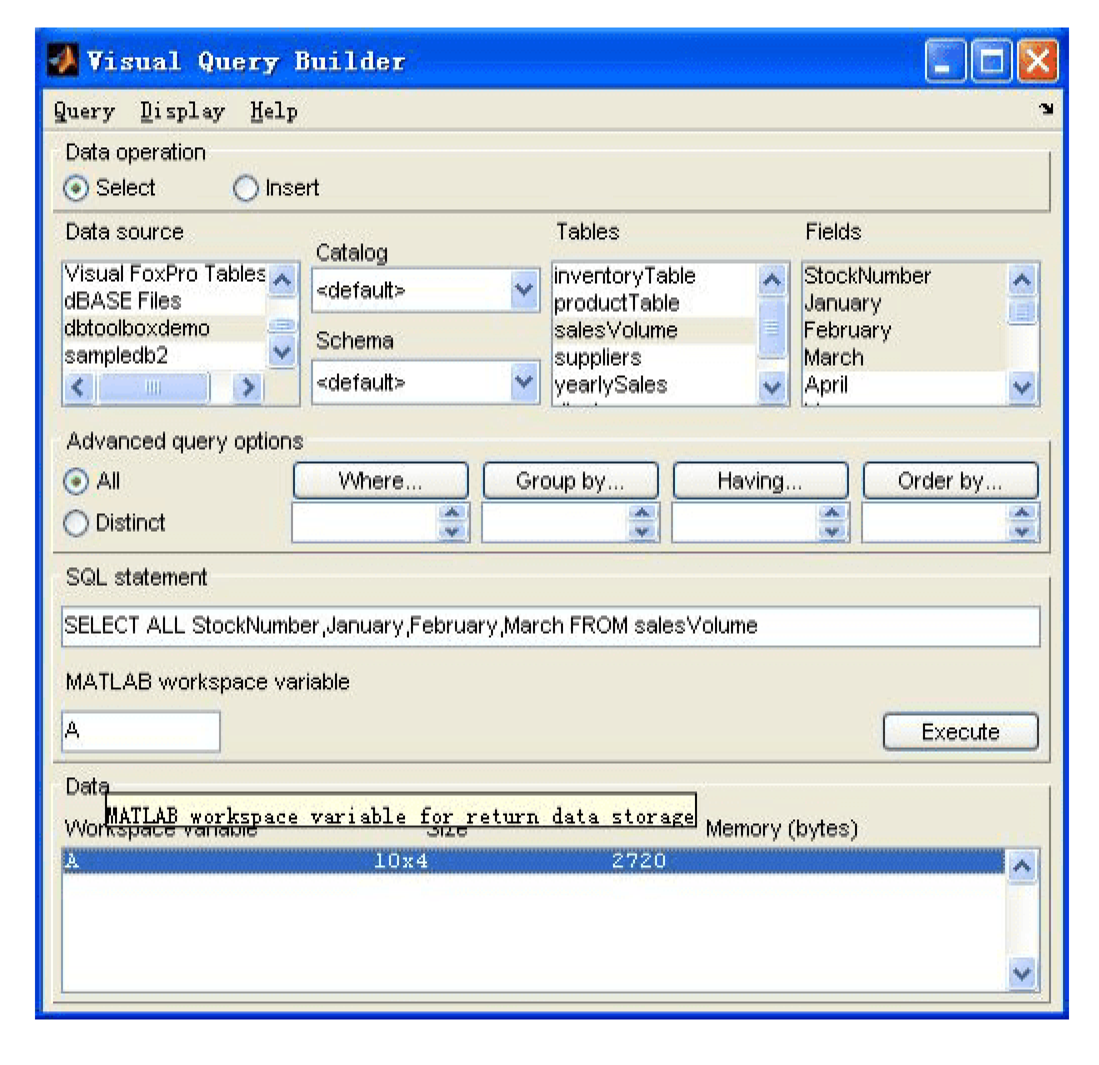


图 10.23　A 的信息显示在 Data 域

Array Editor – A

Stack: Base

	1	2	3	4	5
1	125970	1400	1100	981	
2	212569	2400	1721	1414	
3	389123	1800	1200	890	
4	400314	3000	2400	1800	
5	400339	4300	NaN	2600	
6	400345	5000	3500	2800	
7	400455	1200	900	800	
8	400876	3000	2400	1500	
9	400999	3000	1500	1000	
10	888652	NaN	900	821	
11					
12					
13					
14					
15					
16					
17					

图 10.24　A 的信息显示在数组编辑器中

5. 保存和编辑查询

（1）保存查询。

① 从 Query 菜单项中选择 Save 命令，打开 Save SQL Statement 对话框。

② 在 File name 域中输入文件名（假设为 qfile.qry），单击 Save 按钮，文件被保存在 MATLAB 根目录中的 work 文件夹中。此时只保存了查询本身，并不保存工作区变量和查询优选权。

（2）使用保存的查询。

① 从 Query 菜单项中选择 Load 命令，打开 Load SQL Statement 对话框。

② 选择要加载的查询名（qfile.qry），单击 Open 按钮，VQB 的多个域中立即反映出被查询

的值。

③ 给查询结果指定一个变量名，单击 Execute 按钮。

（3）编辑查询。

已经建立的或加载的查询，都可以修改，然后再执行或保存。

也能在 VQB 中直接修改 SQL 语句。

（4）消除 Data 域中的变量。

Data 域中包括为查询结果定义的变量和在命令窗口中定义的变量。只有在执行了正确的查询后，在命令窗口中定义的变量才出现在 Data 域中。

在命令窗口执行 clear 命令时，可清除变量。被消除的变量不会自行从 Data 域消失，也要在执行了正确的查询后才消失。

6. 为 NULLS、数据格式和错误处理指定优先权

所谓优先权，是指在对 NULLS、数据格式和错误处理时，被指定了选择格式具有被优先使用的权利。

（1）从 Query 菜单项中选择 Preferences 命令，打开 Database Toolbox Preferences 对话框，此时图中显示的都是默认值。

（2）改变当前的设置值，然后单击 OK 按钮，即可改变优先权。

如可以将 Data return format 设置为 numeric（默认值为 cellarray）。这两种格式读取的数据，数组格式占的内存大，处理速度慢，显示在命令窗口的数据带有方括号；而数字格式占的内存小，处理速度快，显示在命令窗口中的是数据本身。

将 Real NULL numbers as 设为 0 时（默认值为 NaN），当读取的数据库中出现 NULL 数值时，现在就用 0 表示。

将 Error handling 设为 report 时（默认值为 store），当执行查询时产生的任何错误都立即显示在命令窗口中。

对于上述的设置，也可以用 setdbprefs 函数实现。

图 10.25 所示的即为 A 在 MATLAB 工作窗口显示的格式。

Array Editor - A

	Cut	2	3	4
1	125970	1400	1100	981
2	212569	2400	1721	1414
3	389123	1800	1200	890
4	400314	3000	2400	1800
5	400339	4300	0	2600
6	400345	5000	3500	2800
7	400455	1200	900	800
8	400876	3000	2400	1500
9	400999	3000	1500	1000
10	888652	0	900	821
11				

图 10.25　A 显示的格式（数字矩阵，NaN 用 0 表示）

7. 浏览查询结果

在 VQB 中执行查询后，就可以在命令窗口或数组编辑器中看到查询的结果。但事实上，VQB 还可以用其他方式来处理和查看结果。

（1）数据的关系表。

从 VQB 的 Display 菜单项中选择 Data 命令，便可以生成图 10.26 所示的数据表。

从数据表中还可以显示出数据之间的关系。如单击 January 字段的 3000，则相关的数用黑体

显示，并且用点线连接。它表示 January 的销售额 3000 单位对应的 StockNumber（货号）有 400314、400876 和 400999。根据连线可以看出这三个货号的商品在 February 和 March 的销售额。

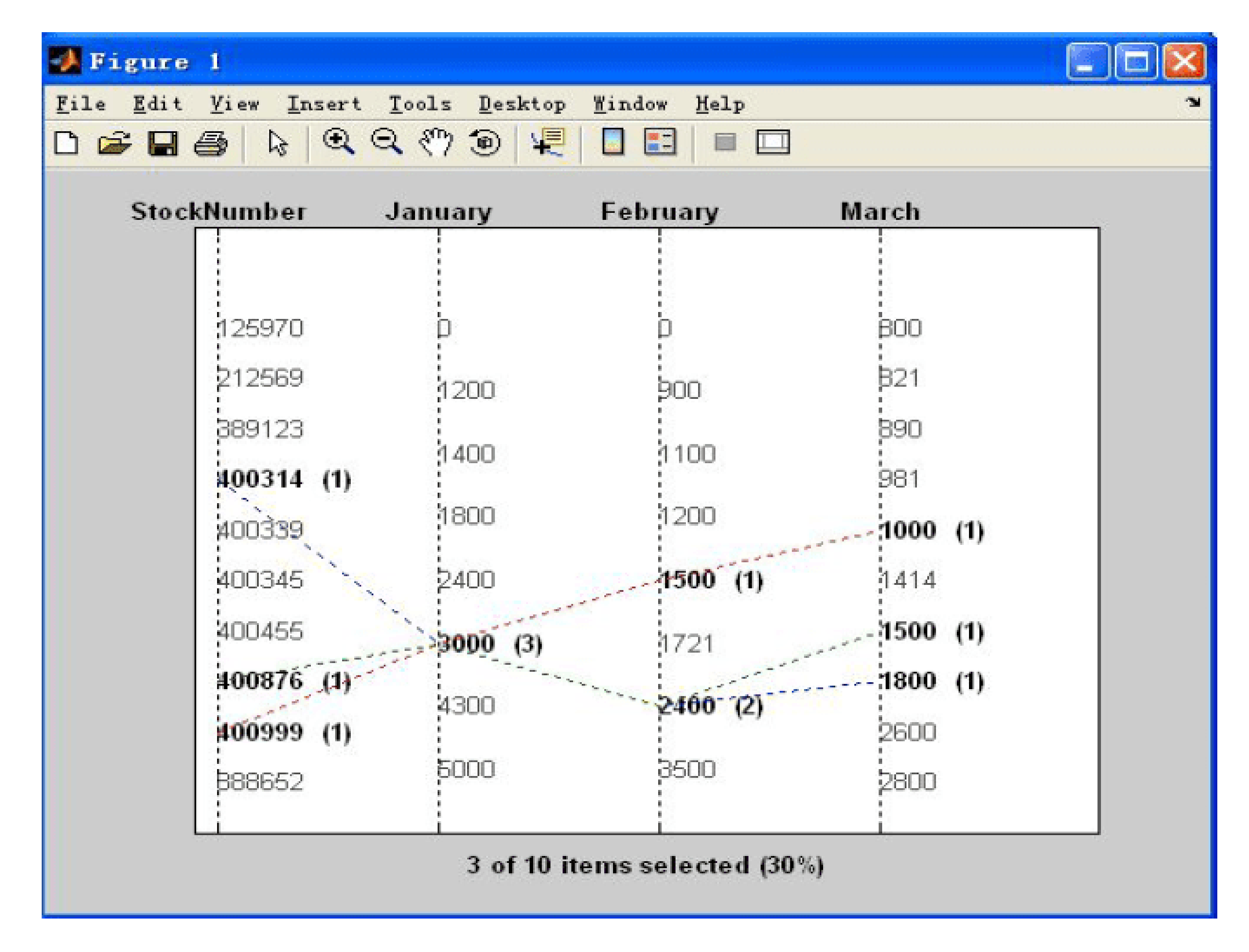


图 10.26　数据表及其数据之间的关系

（2）图表显示结果。

从 VQB 的 Display 菜单项中选择 Chart 命令，打开图 10.27 所示的对话框。

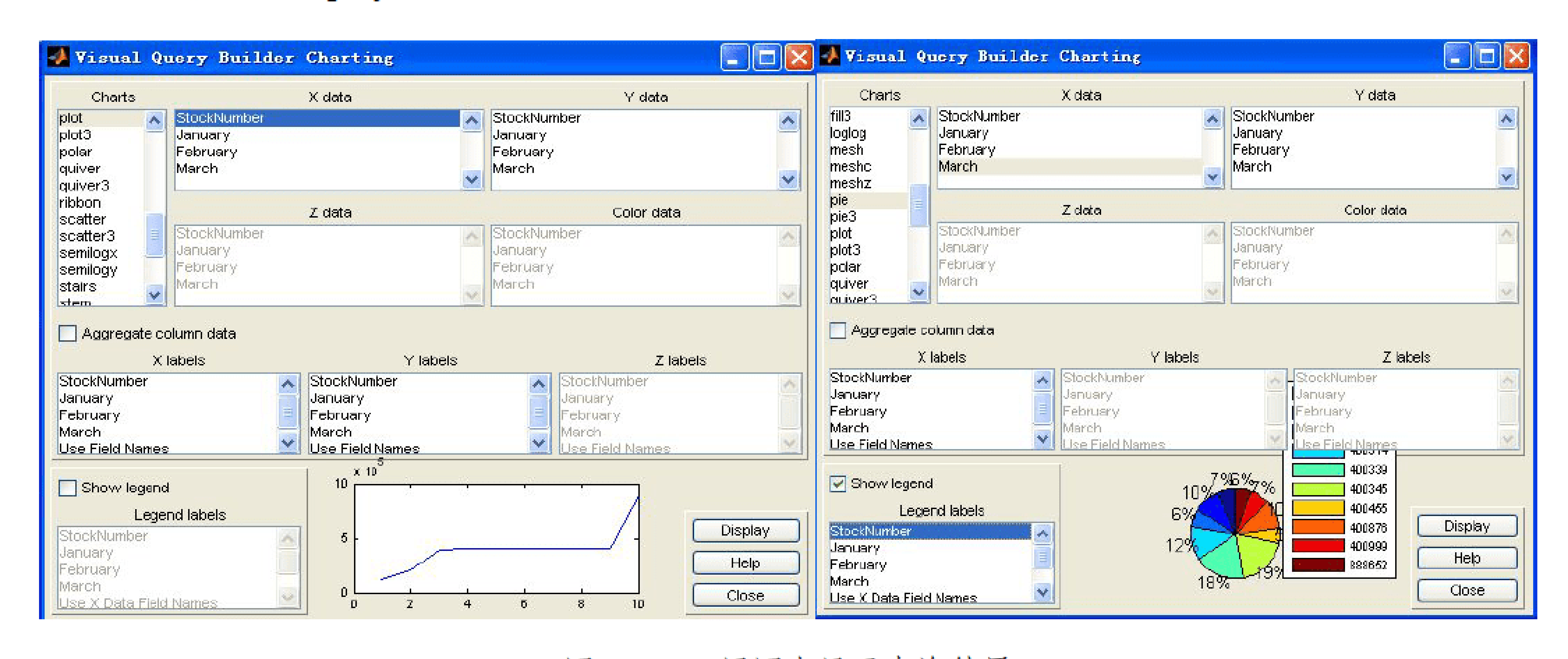


图 10.27　用图表显示查询结果

从图中选择合适的选项，便可以得到不同类型的图形，还可以单击图中的 Display 按钮，可以将图形显示在图表窗口中。

图例的大小和位置可以改变。单击图例，在弹出的快捷菜单中选择 Properties 命令，则图例四周出现可拉动的黑块，拉动它们即能改变大小，直接拖曳图例，则可以移动其位置。

选择 Insert 菜单，可以对图例加注标注。

（3）报表显示在 Web 浏览器上。

从 VQB 的 Display 菜单项中选择 Report 命令，查询结果报表便可以出现在 Web 浏览器上。在这个报表中，一行是一个记录。

从 VQB 取来的字段值若没有字段名，可以通过在 MATLAB 工作窗口中修改变量 A 而得到表头名，如图 10.28 所示。

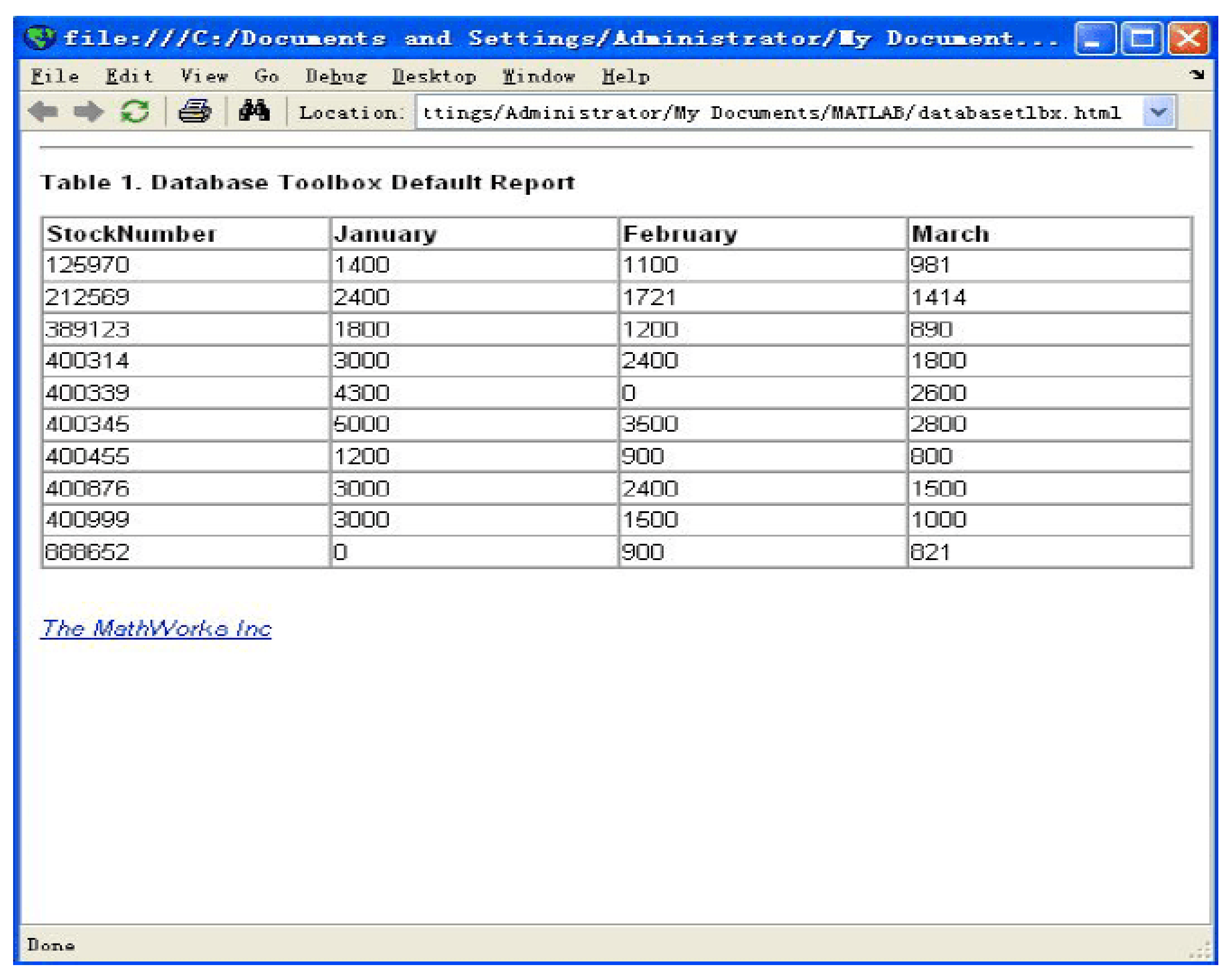


Table 1. Database Toolbox Default Report

StockNumber	January	February	March
125970	1400	1100	981
212569	2400	1721	1414
389123	1800	1200	890
400314	3000	2400	1800
400339	4300	0	2600
400345	5000	3500	2800
400455	1200	900	800
400876	3000	2400	1500
400999	3000	1500	1000
888652	0	900	821

The MathWorks Inc

Done

图 10.28　A 的结果显示在 Web 浏览器中

```
>> A={{'stocknumber','January','February','March'};A};
```

然后在报表生成器中，修改 Header/Footer Options 中的 Number of header rows 为 1，则输出报表时字段名便会出现在表头中。

使用 Web 浏览器能够将这个报表保存为 HTML 页，以供今后查看，也可以使用浏览器的打印功能打印报表。

（4）利用报表生成器定制报表。

从 VQB 的 Display 菜单项中选择 Report Generator 命令，打开报表生成器窗口，如图 10.29 所示。

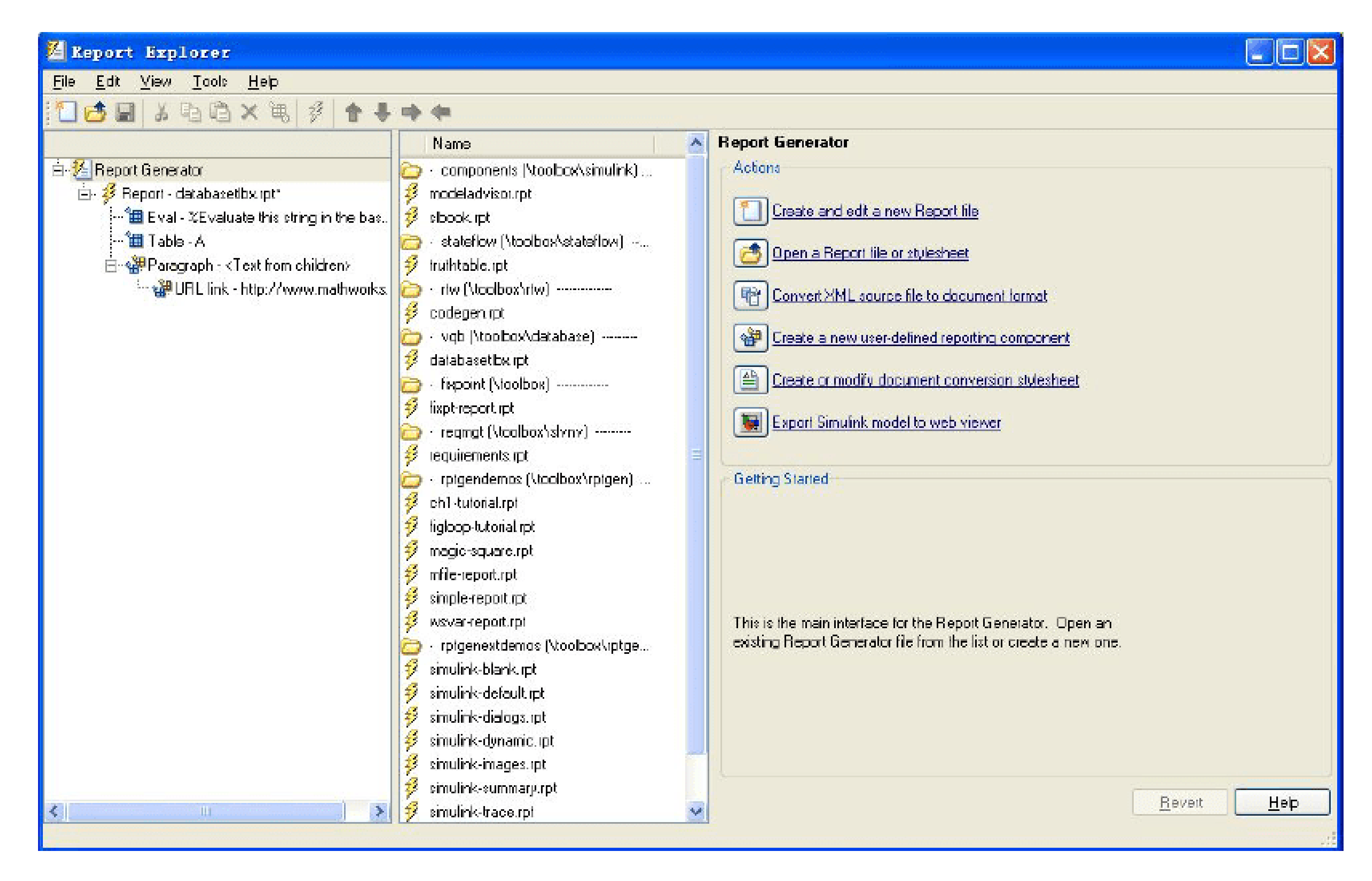


图 10.29　报表生成器

在窗口中间的 Name 列表框中选择 databasetlbx.rpt（这是 VQB 自带的一个报表模板），然后单击 Open report 按钮，打开报表选项窗口。

从窗口左边的 Report Generator 列表框中选择 Table-ans 选项，然后将 Workspace variable name 文本框中的 ans 改成 A（结果变量），将 Number of header rows 中的“1”改成“0”（即报表没有报头），在 File 菜单中选择 Report 命令，则生成的报表就显示在 Web 浏览器中。同样可以利用前述的方法加上表头。

8. 精细查询

精细查询即为高级查询，包括只提取字段中唯一的值，读取符合指定条件的信息，按指定顺序排列查询结果，为多个表中的值建立子查询、联合查询等。

（1）读取的字段值避免重复。

在 VQB 窗口中，Advanced query options 下有一对单选按钮：All 和 Distinct。选择 Distinct 单选按钮，表示读取的字段值没有相同的，即相同值只取一个；而选择 All 单选按钮，则读取字段的所有值。

（2）读取符合指定条件的信息。

在 Advanced query options 中的 Where 域，用来给出条件，以形成 SQL 语句的条件子句。

单击 Where 按钮，即可打开 Where Clauses 对话框，便可以指定选择条件。

在 Condition 下面选择 Relation，在其下拉列表框中选择“>”，再在右边文本框中写入 400000，单击 Apply 按钮，则条件 StockNumber>400000 就出现在 Current clauses 域中。

从 Current clauses 中选择 StockNumber>400000；单击 Edit 按钮，在 Operator 中选择 AND 选项，单击 Apply 按钮，则 Current clauses 改变为 StockNumber>400000 AND，（也可以直接在编辑 StockNumber>400000 时后选择 Operator 中选择 AND）。

以同样的方法，加上另一个条件。最后单击 OK 按钮，关闭 Where Clauses 对话框。此时在 VQB 的 SQL 域中，有一条完整的语句，如图 10.30 所示。

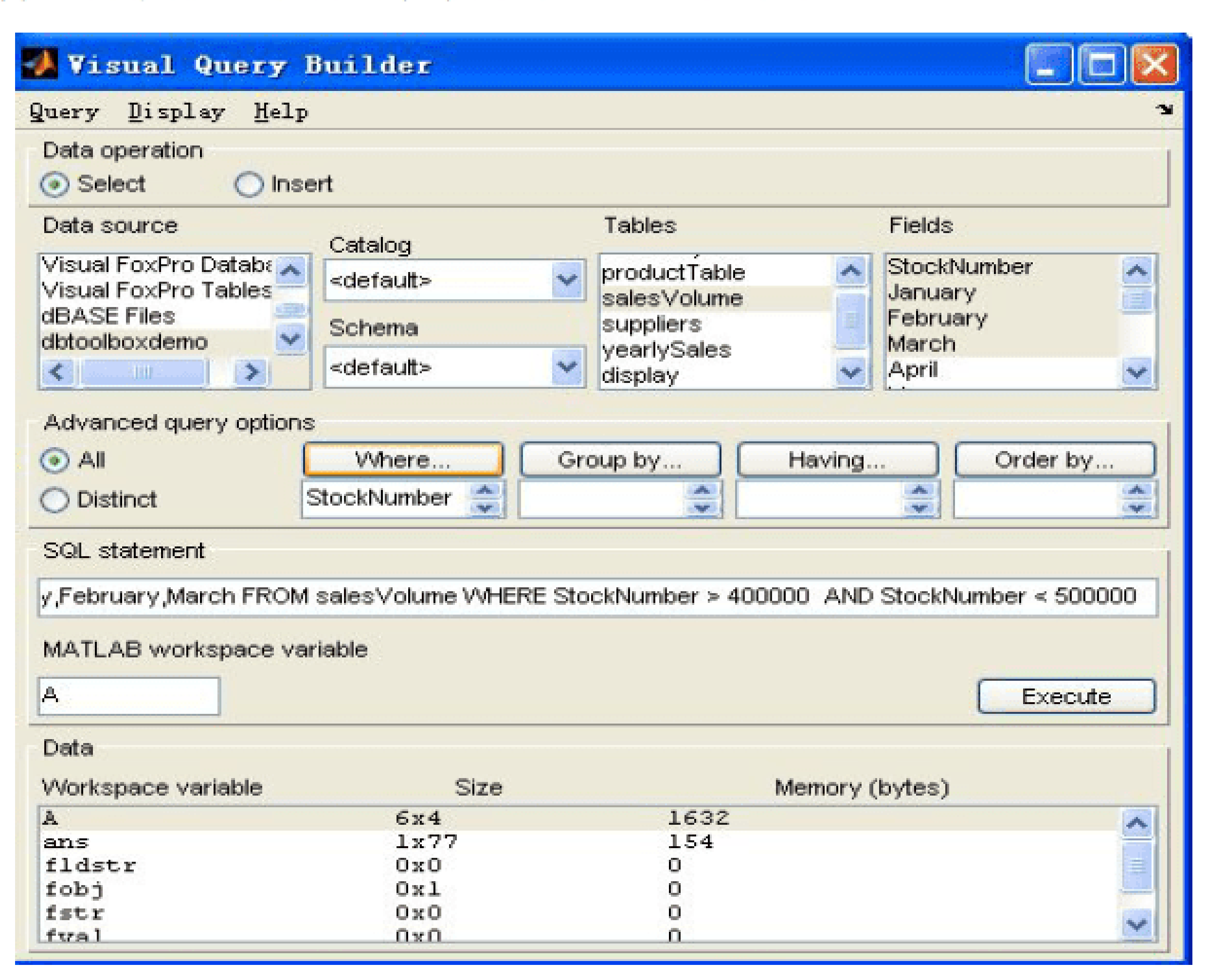


图 10.30 指定查询条件的 VQB

单击 Execute 按钮，便可得到查询结果。

（3）给指定条件分组。

这是前面的 Where 子句的继续。按照前述的方法编辑好条件语句后，选择需合并的条件，再单击 group 按钮，即可将子句中的几个条件合并为一个，使其成为一个整体条件。要注意逻辑运筹符的优先次序，这样才能正确编辑查询条件语句。

（4）按指定顺序排列结果。

查询得到的结果，是按记录在数据库中的顺序排列的，数据随机出现。在实际应用中，为了对某些数据做直观的比较，需要对数据重新排序。

在 VQB 中，Advanced query options 中的 Order by 提供了重新排列数据的功能。在打开的 Order by Clauses 对话框中的 Sort key number 是排列项的顺序。

（5）用多个表的值建立子查询。

Where 域不仅可以建立条件查询，还可以建立子查询。即利用其他表中相关的值作为条件，进一步限制查询，这是 SQL 语句的嵌套。使用 VQB，只能建立一个子查询，而利用数据库函数可以建立多个子查询。

建立子查询，需要利用 Subquery 域，建立方法与前类似，按提示进行，不再赘述。

（6）联合查询—结果来自多个表的查询。

联合查询，是指在 VQB 中建立查询，要选择几个表，从每个表中抽取所需的信息，组成一个结果。

联合查询与子查询有点不同。联合查询在建立查询时选择多个表，这些表不一定包含共享字段，不建立子查询。而子查询是建立查询和子查询时分别选择不同的表，这些表必须包含共享字段。

可以在 VQB 窗口中的 Table 域中同时选择所指定的表，如同时选择 productTable 表和 SalesVolume 表，由此 Fields 域列出了字段名，可以看出，此时字段名前加了表名。

再在 Fields 域中同时选择所需的表，如 productTable.productDescription、salesVolume.January、salesVolume.February 和 salesVolume.March。

再利用 Where 域，指定查询条件，便可得到查询结果，如图 10.31 所示。

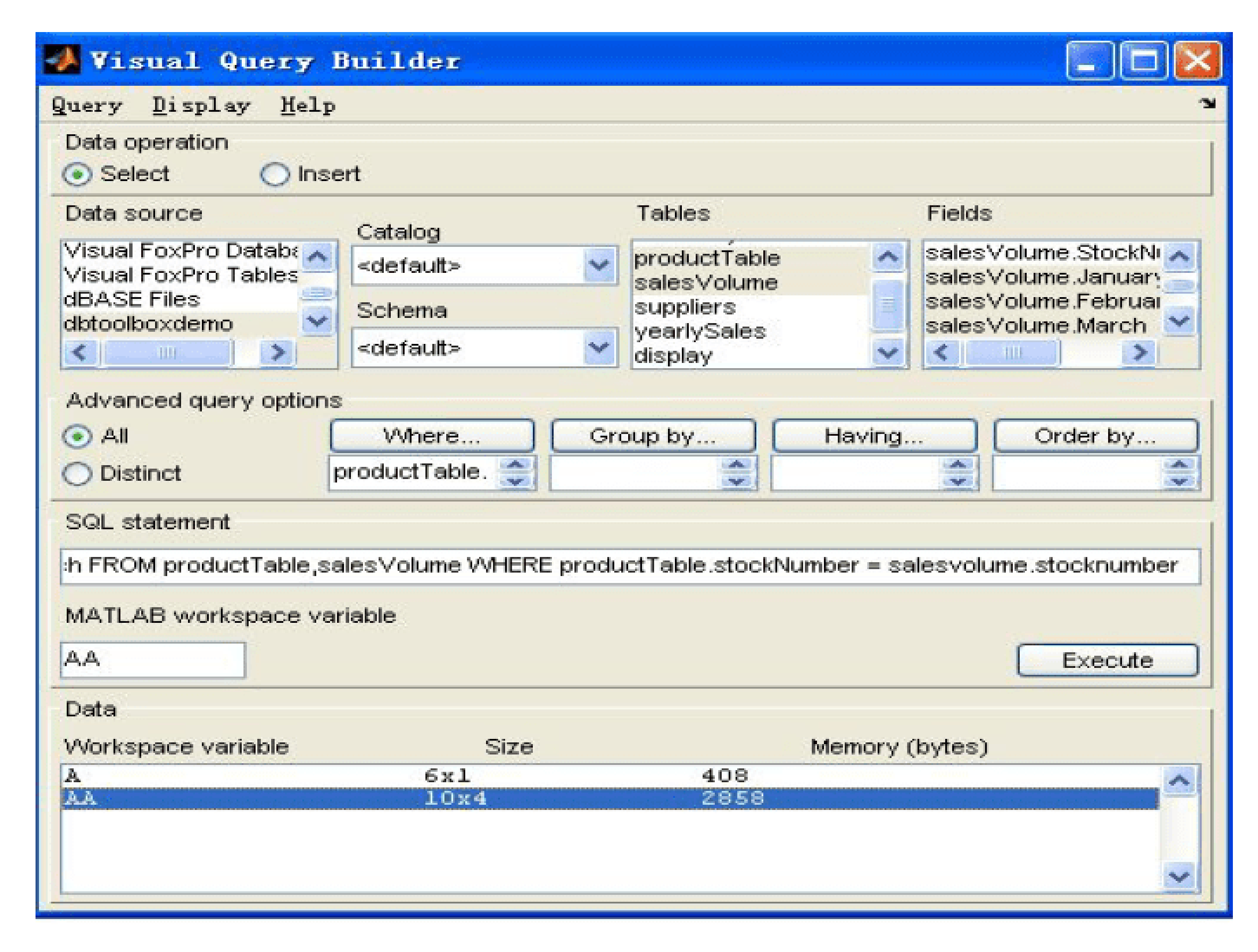


图 10.31　VQB 查询表

（7）用 VQB 输出数据。

将数据从 MATLAB 输出到另一数据库，可以在 VQB 中建立输出数据的查询并执行它。

这种输出只能写新数据行到数据库，而不能用新数据替换已存在的数据，也即只能增加而不能修改。要修改原来的数据，用 update 函数。

此时，首先在 VQB 界面中选择插入内容所在的数据源，在 Data Operation 域选择 Insert，在 Table 和 Fields 中选择插入内容所在的表和对应的域，然后命名变量名称，单击 Execute，就可将数据插入到数据库，并在 matlab command 文本框中显示生成的查询语句。

9. MATLAB 的逻辑型数（Boolean Data）

在数据库应用中，免不了用到逻辑型数，即某个字段的值表示两种状态之一：是或否（真或假）。在 MATLAB 中，用 Boolean 表示逻辑型数。

VQB 可以输入或输出此类型数，当然它只能存储在单元数组或结构数组中，它有两个值：0（表示假），1（表示真）。

（1）输入 Boolean 类型数据。

在所选择的数据库中，如果某表中的某字段是用复选框的打钩表示或用 false 或 true 表示，或用 0 或 1 表示某种状态等等，这些都说明它们为逻辑类型的数据。在这种情况下，就可以按照前述的方法进行查询，其查询结果以二值矩阵表示。

（2）输出 Boolean 类型数据。

此操作可以在数据库中插入 Boolean 类型数据。此时应先在 MATLAB 窗口里，建立一个如下形式的命令作为输出的结构：

```
A1.ProductName{1}='巴西咖啡';
A1.Discontinued{1}=logical(1);
```

打开 VQB，在 Data operation 域中，选择 Insert，在 Table、Fields 中选择数据源中所要查询的表及域，在 matlab workspace variable 中指定变量 A1，单击 Execute 按钮，就可以将所建立的输出结构加入到相应的数据库中。

10.6.3　数据的存取类型

在数据仓库中，经常会遇到较为复杂结构的数据，为了能更好地存取这些数据，可以应用以下方法。

1. 结构数组

结构是一种用字段容纳数据的 MATLAB 数组，结构的字段能包含任何类型的数据，它的操作与一般的数组操作方法类似。

（1）结构数组的建立。

建立结构数组有两种方法：

① 使用赋值语句。

直接对单个字段指定数据，建立一个简单的 1×1 的结构数组。

```
>> patient.name='John Doe';patient.billing=127;
>> patient.test=[79 75 73;180 178 177.5;2202 210 205];
>> patient
patient =name: 'John Doe'
  billing: 127
       test: [3×3 double]
```

要对此数组进行扩展，只要在结构名后加下标（即数组的下标），再做一次赋值即可。如果有未指定的字段，MATLAB 用空矩阵填充。

使用 fieldnames 函数，可以得到结构所含的字段，字段名以字串单元数组的形式出现。

② 利用 struct 函数建立结构数组。

struct 函数的调用格式：

```
s=struct('field1',values1,'field2',values2,…),其中 fieldi 表示字段名,valuesi
表示对应于的字段值,必须是同样大小的单元数组或标题
>> s=struct('string',{{'hello','yes'}},'lengths',[5 3]);  %注意第二个字段值
>> s=string: {'hello'  'yes'}
    lengths: [5 3]
```

struct 函数能够在内存中为结构数组预先保留位置，即在使用前预分配内存单元。

（2）使用动态字段名。

存取结构数组数据的常用方法并不适合较为复杂程序的应用，因为事先不能将字段名一一列出，此时可应用动态字段名，即用变量表示字段名，在程序运行时，使真正字段名代替变量，以便存取值，其语法形式为：structName.(expression)。在这些形式后面加标准索引，就可以存取某字段的某个或某些元素值。

例下面的例子：

```
function testn=gettest(patient,test)
  testn=patient.(test)(2,1:3)
>> patient=struct('name',{'John','Ann Lane'},'test1',{[79 75 74;73 87 90],…
           [99 93 86;91 72 95]},'test2',{[66 67 68;69 71 82],[83 74 90;61 82 93]});
>> gettest(patient(1),'test1')
 testn=73 87 90
```

（3）增加和删除结构字段。

任何已经存在的结构数组，都能增加字段，而且只要为其中的任何一个结构增加就可以，MATLAB 会自动把增加的字段扩展到其他结构。

利用 rmfield 函数可以从结构中删除一个或多个某字段：s=rmfield(s,'field')

（4）用结构数组组织数组。

考虑一个 128×128 的 RGB 图像，目前图像的数据按 Red、Green 和 Blue 分别存储于 3 个独

立的数组中。

可以按两种方式将 RGB 图像组织为结构数组。

① 平面组织。

把上述的 3 个数组变为结构数组 A 的 3 个字段，每个字段是完整的一个图像平面：

```
>>A.r=red;A.g=green;A.b=blue;
```

② 单元素组织。

单元素组织，就是把 RGB 图像 3 个独立数组的单个数组，作为结构中每个字段的值。这种组织的优势，显然在于存取图像的子集。

从 RGB 图像 3 个独立的数组，建立一个 128×128 个结构的数组，需要用循环语句完成：

```
for i=1:size(RED,1)
  for j=1:size(RED,2)
    B(i,j).r=RED(i,j);B(i,j).g=GREEN(i,j);B(i,j).b=BLUE(i,j);
  end
end
```

这两种结构各有优缺点。平面结构更适合于一次操作所有的字段；而单元素组织更容易存取单个客户的所有信息。

（5）嵌套结构。

在结构的字段中包含另一个结构，甚至是一个结构数组，这就是嵌套结构。

一旦建立了结构，就可以用 struct 函数，或者用直接赋值语句，在已有的结构字段中建立嵌套结构。

```
>> a=struct('data',[3 4 7;2 4 6],'nest',struct('testnum','test1','xdata',[4
2 8],'ydata',[7 1 6]))
   a = data: [2×3 double]
   nest: [1x1 struct]
```

存取嵌套结构中的数据，索引要复杂一些，需使用点操作符将嵌套的字段名一一罗列在索引中。字段名的前后次序，表示字段嵌套的层次。索引表示式中的第 1 个文本串，是结构数组的名称，后面是包括其他结构的字段名。

```
>> a.nest=testnum: 'test1'
 xdata: [4 2 8]
 ydata: [7 1 6]
>> a.nest.xdata=4  2   8
```

（6）多维结构数组。

多组结构数组是方形结构数组的扩展。与普通多维结构数组一样，多维结构数组可以使用直接赋值语句或 cat 函数建立。

```
>> patient(1,1,1).name='John';patient(1,1,1).biling=127.0;patient(1,1,1).test=[1  2
```

```
3;3 4 5];
>> patient(1,1,2).name='Al Smith';patient(1,1,2).biling=207.0;patient(1,1,2).test=[4
6 8;2 1 7];
>> patient(1,2,1).name='Ann Lane';patient(1,2,1).biling=130.0;patient(1,2,1).test=[3
9 2;3 2 7];
>>  patient(1,2,2).name='Dora';patient(1,2,2).biling=120.0;patient(1,2,2).test=[1  5
3;4 3 5];
```

以上数组中的第 3 个数字表示页。

2. 单元数组

单元数组是在一个数组中包含多个单元（cell），每个单元作为一个独立的存储单元存储。单元中的数据可以是数组、字串、向量或标题。

（1）建立单元数组。

可以分别使用函数及赋值方法建立单元数组。

① 使用赋值语句建立单元数组。

为单元赋值，可采用两种索引方式。

I.单元索引

```
>>A(1,1)={[1 2 3;0 9 8]};A(1,2)={'abcdefg'};A(2,1)={8+9i};A(2,2)={-pi:pi/10:pi};
>>A=   [2×3 double]        'abcdefg'
    [8.0000 + 9.0000i]    [1×21 double]
```

II.内容索引

```
>>A{1,1}=[1 2 3;0 9 8];A{1,2}='abcdefg';A{2,1}=8+9i;A{2,2}=-pi:pi/10:pi;
```

如果被赋值的单元在当前数组的维数之外，MATLAB 将自动扩展这个数组，以包含指定的下标，并且用空矩阵填充插入的单元。

```
>>A(3,3)={5};
>>A=[2×3 double]          'abcdefg'        []
    [8.0000 + 9.0000i]    [1×21 double]    []
        []                    []              [5]
```

② 利用 cell 函数作为单元数组预留空间。

用 cell 函数能够预先分配指定大小的空单元数组。其基本调用格式为：

```
A=cell(n,m);
```

③ 显示单元数组内容。

可以分别用直接求索引、函数 celldisp 显示单元数组的内容。

（2）嵌套单元数组。

可以 cell 函数、嵌套的方括号和直接赋值的方法建立嵌套的单元数组。

```
>> A=cell(1,2);A(1,2)={cell(2,2)};
>> A=[]    {2×2 cell}
```

（3）结构的单元数组。

用单元数组存储具有不同字段的结构，是单元数组与结构的结合。

```
>> mystr=cell(1,2);mystr{1}.xdata=[1 2 3;4 5 6];mystr{1}.ydata=[3 5 3;6 7 3];
>> mystr{2}.label='2014-10-1'; mystr{2}.obj=[1 3];
>> mystr=[1×1 struct]    [1×1 struct]
```

（4）多元单元数组。

多元单元数组是二维单元数组的扩展，建立方法与建立多元数字数组一样。

3. 多维数组、单元数组或结构数组

在 MATLAB 中，把大于二维的数组（单元数组/结构数组）、称作多维数组（单元数组/结构数组）。能在矩阵上执行的大部分操作都可以用于多维数组（单元数组/结构数组）。

存取多维数组（单元数组/结构数组）的元素需要更多的下标。例如对于三维数组，使用 3 个下标：第 1 个下标表示行索引，第 2 个下标表示列索引，第 3 个索引表示页索引。而对于更多维的数组，一般很难想象它的空间模样。

```
>> rand(2,1,3)          %三维数组
ans(:,:,1) =0.8147      %第 1 页,在此页为 2×1 数组
    0.9058
ans(:,:,2) = 0.1270     %第 2 页
    0.9134
ans(:,:,3) = 0.6324     %第 3 页
    0.0975
```

（1）建立多维数组。

建立矩阵（单元数组/结构数组）的方法都可以用来建立多维数组（单元数组/结构数组）。例如使用索引来扩展数组、使用 rand、ones、zeros 和 repmat 等函数。

（2）多维数组（单元数组/结构数组）的索引。

适用于矩阵的一些概念可以推及到多维数组（单元数组/结构数组）。诸如存取单个元素使用整数下标；使用冒号用于索引表达式以存取子集或数组的整行、整列或全部页；使用线性索引将每一页作为一个矩阵，逐页排列，形成一个更大的列向量。

（3）多维数组（单元数组/结构数组）作为运算对象。

MATLAB 的许多计算和数学函数接受多维数组（单元数组/结构数组）作为参数。这些函数把多维数组（单元数组/结构数组）指定的维作为运算对象，即它们运算单个元素、向量或矩阵；或者是把提取单个维的数据再作各种运算。

（4）用多维数组（单元数组/结构数组）组织数据。

在 MATLAB 中，用多维数组（单元数组/结构数组）存储数据有两种方法：一是平面形式，

即用二维数组存储数据，然后把数据作为矩阵处理；二是立体形式，即用三维或更高维的数组存储数据，然后或者处理其中的页，或者处理数据子集。

10.6.4 数据输入和输出

MATLAB 提供了许多输入和输出数据的方法。这里的输入是指磁盘文件或剪贴板中加载数据到 MATLAB 工作区；输出是指保存工作区的变量到磁盘文件。输入或输出数据选择哪一种方法，主要取决数据的格式：文本、二进制或标准格式如 HDF 等。

MATLAB 中有许多用于数据输入的函数，可以根据数据的不同格式，选用合适的输入函数。另外，还可以用工具箱（或从 import data 菜单入手）执行专门的输入任务。

1. 保存和加载 MAT 文件

MAT 文件是双精度、二进制、MATLAB 格式文件。

使用 save 函数可以把工作区的变量输出到二进制或 ASCII 文件，如果缺省文件名，MATLAB 指定为 matlab.mat，其调用格式为：

```
save filename var1 var2 …varN
```

表示将各个变量（可以使用通配符*）保存到文件 filename 中。

使用此函数，还可以保存整个结构，也可以把结构的每个字段作为一个变量保存，或把指定的每个字段保存为单独变量。

如果在 save 函数中加入-append 选项，则可以将新的变量添加到已经存在的 MAT 文件中。

在使用 save 函数保存文件时，缺省值是压缩数据，如果要禁止压缩，需在 save 命令中加-v6 选项。

使用 load 函数可以从二进制或 ASCII 文件加载变量到 MATLAB 工作区，如果缺省文件名，MATLAB 使用 matlab.mat，其调用格式为：

```
load filename var1 var2 …varN
```

表示将文件上指定的变量加载到工作区。

2. 输入文本数据

可以用多种方法输入文本数据，但究竟选用哪一个，取决于文件的格式。文本文件必须是格式化的。它的行与列有整齐的形式，数据之间用定界符分隔，定界符可以是空格、逗号、分号、制表符或其他字符。单个项可以是字母、数字或字母数字的混合。

文本文件也能够包含一个或多个头行。文本头有 3 种，即文本头（即文件标题）、行头（行标题）和列头（列标题），它们分别用来标识文件、行和列。

输入文本数据的函数有：csvread、dlmread、fscanf、load、textread 和 textscan 等。这些函数的应用格式可参见其帮助文件。

3. 输入带有文本头的数据

输入带有文本头的数据文件可以用 textscan 函数。在应用此函数时，首先必须用 fopen 函数

打开文件。fopen 为 textscan 提供了它需要的文件标识符 fid。完成读操作后，应当使用 fclose 函数关闭文件。

textscan 函数的调用格式为

C=textscan(fid,'format',param,value,...)

其中其支持的转换定义符如表 10.9 所示。

表 10.9　textscan 支持的转换定义符（即'format'）

类　别	定 义 符	实际意义
数字字段格式	%n,%d,%f 和其他类似的定义符（如%d16）	读到第一个界定符
	%Nn,%Nd,%Nu,%Nf 和其他类似的定义符	读 N 位（包括小数点），或直到第一个定界符为止。先遇定界符，在定界符结束读，否则读满 N 位
	由%N.Df 开头的定义符	读 N 位（包括小数点），或直到第一个定界符为止。先遇定界符，在定界符结束读，否则读满 N 位
字串字段格式	%s 或%q	读到第一个定界符
	%Ns 或%Nq	读 N 个字符，或直到第一个定界符。先遇到定界符，在定界符结束读
	%[abc]	读，直到遇见第一个未指定在括号中的字符为止
	%N[abc]	读 N 个字符，或直到第一个未指定在括号中的字符为止
	%[ˆabc]	一直读到第一个指定在括号中的字符为止
	%N[ˆabc]	读 N 个字符，或直到第一个指定在括号中的字符为止
字符字段格式	%c	读一个字符
	%Nc	读 N 个字符，包括定界符

对于含有字母和数字混合数据的文件，使用 textscan 和 textread 函数都可以输入。

textread 函数的调用格式

[A,B,C,…]=textread('filename','format')

或

[…]=textread(…,'param','value')

其中其支持的转换定义符如表 10.10 所示，参数/值对表如表 10.11 所示。

表 10.10　textread 函数格式转换定义符

定 义 符	实际意义
被跳过的字符	读文件时跳过与此相匹配的字符
%d	读带符号的整数
%u	读无符号的整数
%f	读浮点数
%s	读由空线间隔或定界符分隔的字串
%q	读由双引号中的字串，忽略引号

续表

定 义 符	实际意义
%c	读字符，包括空线间隔
%[⋯]	读与括号中字符相匹配的字符，直到遇上第一个不匹配的字符为止
%[^⋯]	读与括号中字符不匹配的字符
%*⋯代替%	读文件时忽略被*指定的字符
%w 代替%	读字段，字段长度由 w 限定。读浮点数，可写成%w.pf，这时的 w 表示字段长度，而 p 是精度

表 10.11　textread 函数参数/值对

参　数	值	作　用
	''	空格
	\b	退格
	\n	换行
	\r	回车
	\t	水平 tab
bufsize	正整数	定义最大字串长度，用字节表示
commentstyle	matlab	%是 MATLAB 的注解符，读数据时忽略%后的字符
commentstyle	shell	读数据时忽略＃后的字符
commentstyle	c	读数据时忽略/*和*/之间的字符
commentstyle	c++	读数据时忽略//后的字符
delimiter	1 个或多个字符	作为字段界定符，没有默认值
emptyvalue	双精度值	为空单元（空字符）指定值，默认是 0
endofline	单个字符或'\r\n'	表示行结束的字符，默认用文件上的
expchars	指数符号	默认是 eEdD
headerlines	正整数	读文件时忽略的文件头的行数
whilespace	''，\b,\n,\r,\t 中的一个	作为空线间隔的字串，默认是'\b\t'

textscan 和 textread 函数之间有所区别。textscan 有更好的性能，特别适合读大文件。但它在使用时，首先要打开文件，一旦打开文件，可以从文件的任何位置读，并且只要不关闭文件，可以从上次操作的中断点继续读数据，而且只输出一个单元数组，不必给每个被读字段指定一个输出参数；而 textread 每次只能从文件的开头读，并且不需要用其他函数打开文件，不能继续读。

4. 输入/输出电子表数据

（1）得到文件的有关信息。

可以用 xlsfinfo 函数确定文件是否为可读的 Excel 电子表。

（2）将数据输出到文件。

可以用 xlswrite 函数将 MATLAB 工作区的数据输出到文件的任何工作表和工作表的任何位

置，其调用格式：

```
xlswrite('filename',M,sheet)
```

其中：M 为工作区中被输入的矩阵，filename 为电子表文件名，sheet 为指定的工作表。如果 sheet 不存在，则一个新工作表被加到工作表集的尾部；如果 sheet 是一个大于工作表数目的索引，则附加一些空工作表，直到工作簿中的表数目等于指定的 sheet 为止。

（3）从文件输入数据。

xlsread 函数可以从 Excel 电子表文件将一个矩阵输入到 MATLAB 工作区。可以从文件的任何候选工作表、表的任何位置输入数据。也能够由 xlsread 打开 Excel 窗口，然后交互地选择工作表和被读数据的位置。

xlsread 函数的调用格式：

```
N=xlsread('filename',-1)                %-1 为在 Excel 窗口打开文件的标志
N=xlsread('filename',sheet)             %读工作表 sheet 上的数据
N=xlsread('filename',sheet,'range')     %读工作表 sheet 上指定区域的数据
```

5. 低级文件输入/输出函数

MATLAB 文件 I/O 函数，可以输入/输出数据。要读或写数据，需要执行下面步骤：

（1）使用 fopen 函数打开文件，返回文件的标识符，标识符将被用在所有其他低级文件 I/O 函数中。

（2）在文件上进行下述操作：

① 使用 fread 函数读二进制数据；

② 使用 fwrite 函数写二进制数据；

③ 使用 fgets/fget1 函数从文本文件逐行读字串；

④ 使用 fscanf 函数读格式化的 ASCII 数据；

⑤ 使用 fprintf 函数写格式化的 ASCII 数据；

（3）使用 fclose 函数关闭文件。

一旦用 fopen 函数打开文件，MATLAB 利用 fseek 函数维持一个文件位置指针，它指示文件上的特别位置，此函数的调用格式：

```
status=fseek(fid,offset,origin)
```

其中 status 为返回值，0 操作成功，1 操作失败；fid 为文件标识符；offset 为移动方向和值，offset>0 向前移动 offset 个字节；offset = 0 不移动；offset<0，向后移动 offset 个字节。

fseek 函数在 fid 指定的文件上依照 origin 给出的参照点，相对移动文件指针 offset 个字节。文件上的字节编号从 0 开始，第 1 个字节为字节 0，以此类推，第 n 个字节为字节 $n-1$。

ftell 函数可以得到文件指针的位置。

读书笔记

第 11 章

模糊集理论

由 L.A.Zadeh 提出的模糊集合理论与模糊逻辑，就是采用精确的方法、公式和模型来度量和处理模糊、信息不完整或不太正确的现象与规律。经过 40 多年的快速发展，模糊理论在诸多学科与工程技术领域得到了很好的应用。

11.1 模糊集合

模糊系统是建立在自然语言基础上的。在自然语言中常采用一些模糊概念如“大约”“左右”“温度偏高”等来表示一些量化指标，如何对这些模糊概念进行分析、推理是模糊集合与模糊逻辑所要解决的问题。

模糊集合是一种边界不分明的集合。对于模糊集合，一个元素可以既属于该集合又不属于该集合，亦此亦彼，边界不分明。建立在模糊集合基础上的模糊逻辑，任何陈述或命题的真实性只是一定程度的真实性。

11.1.1 隶属度函数

如果集合 X 包含了所有的事件 x，A 是其中的一个子集，那么元素 x 与集合 X 的关系可用一个特征函数来描述，这个函数称为隶属度函数 $\mu(x)$。

对于经典的数据集合理论，若 x 包含于 A 中，则 $\mu(x)$取值 1；若 x 不是 A 的元素，则 $\mu(x)$值为 0；而对于模糊集合而言，则允许隶属度函数可取[0，1]上的任何值。模糊集常被归一化到区间[0,1]上，模糊集的隶属度函数既可离散表示，又可以借助于函数式来表示。

隶属函数的表示方法大致有三种：

如 $\underline{A}$ 为模糊集，一般情况下可表示为

$$\underline{A} = \{(u, \mu_{\underline{A}}(u)) \mid u \in U\}$$

如果 U 是有限集或可数集，可表示为

$$\underline{A} = \sum_i \mu_{\underline{A}}(u_i) / u_i$$

此时式子的右端并非代表分式求和，它仅仅是一种符号，分母的位置是论域中的元素，分子位置是相应元素的隶属度。当某一元素的隶属度为 0 时，那一项可以省略。

或表示为向量形式

$$\underline{A} = (\mu_{\underline{A}}(u_1), \mu_{\underline{A}}(u_2), \ldots, \mu_{\underline{A}}(u_n))$$

但要注意，在此形式中，要求集合中各元素的顺序已确定。

如果 U 是无限集，则可以表示为

$$\underline{A} = \int \mu_{\underline{A}}(u) / u$$

同样这里的积分号不是通常的积分含义，只是表示对 u 都指定了相应的隶属度。

隶属度函数可以是任意形状的曲线，取什么形状主要取决使用是否方便、简单、快速和有效，唯一的约束条件是隶属度的值域为[0,1]。

模糊系统中常用的隶属度函数有 11 种，下面介绍常见的几种。

（1）高斯型。该函数有两个特征参数 σ 和 c，其函数形式为

$$\mu(x,\sigma,c)=\mathrm{e}^{-\frac{(x-c)^2}{2\sigma^2}}$$

两个高斯型隶属度函数的组合可形成双侧高斯型隶属度函数。

（2）钟形隶属度函数。该函数有三个特征参数 a、b 和 c，其函数形式为

$$\mu(x,a,b,c)=\frac{1}{1+\left(\frac{x-c}{a}\right)^{2b}}$$

（3）sigmoid 函数型隶属度函数。该函数有两个特征参数 a 和 c，其函数形式为

$$\mu(x,a,b)=\frac{1}{1+\mathrm{e}^{-a(x-c)}}$$

（4）S 型隶属度函数。该函数有两个特征参数 a、b，其函数形式与 sigmoid 函数形式相同，只是参数 a 和 b 的取值不同。

（5）梯型隶属度函数。该函数有四个特征参数 a、b、c 和 d，其函数形式为

$$\mu(x,a,b,c,d)=\begin{cases}0 & x\geqslant a\\ \dfrac{x-a}{b-a} & a\geqslant x\geqslant b\\ 1 & b\geqslant x\geqslant c\\ \dfrac{d-x}{d-c} & c\geqslant x\geqslant d\\ 0 & x\geqslant d\end{cases}$$

隶属度函数是模糊集合赖以建立的基石，要确定恰当的隶属度函数并不容易，迄今仍无一个统一的标准。隶属函数的确定过程本质上是客观的，但又允许有一定的人为技巧。对实际问题建立一个隶属度函数需要充分了解描述的概念，并掌握一定的数学技巧。

在某种场合，隶属度可以采用模糊统计的方法来确定：

① 确定论域 U，如年龄；

② 确定论域中的一个元素 U_0，如年龄为 35 岁的人；

③ 论域中的边界可变的普通集合 A，如“年轻人”，A 联系于一个模糊集合及相应的模糊概念；

④ 判断条件。即对普通集合 A 判断的依据条件。它联系着按模糊概念所进行的划分过程的全部主客观因素，它制约着边界的改变。例如不同的实验者对“年龄为 35 岁的人”的理解。有的认为是年轻人，而有的人则认为不是年轻人。

⑤ 模糊统计实验。其基本要求是在每一次实验下，要对 U_0 是否属于 A 做出一个确切的判断，做 N 次实验，就可以算出对的隶属频率：

$$\text{隶属频率}=\frac{"U_0\in \mathrm{A}"\text{的次数}}{N}$$

其他确定隶属度函数的方法还有二元对比排序法、推进法和专家评分法等。

11.1.2 模糊集运算

与经典的集合理论一样，模糊集也可以通过一定的规则进行运算。实际上模糊集的运算是逐点对隶属度作相应的运算。

1. 交集（逻辑与）

两模糊集的交集 A∩B，为两隶属度 $\mu_A(x)$和 $\mu_B(x)$的最小者

$$f_{A\cap B}(x)=\mu_A(x)\wedge\mu_B(x)=\min|\mu_A(x),\mu_B(x)|$$

2. 合集（逻辑或）

两模糊集的合集 A∪B，为两隶属度 $\mu_A(x)$和 $\mu_B(x)$的最大者

$$f_{A\cup B}(x)=\mu_A(x)\vee\mu_B(x)=\max|\mu_A(x),\mu_B(x)|$$

3. 补集（逻辑非）

$$\mu_{\bar{A}}(x)=1-\mu_A(x)$$

4. 模糊集的基

模糊集的基为隶属度函数的积分或求和

$$card\mathrm{A}=\sum_i\mu(x)$$

$$card\mathrm{A}=\int_x\mu(x)\mathrm{d}x$$

论域 U 上的模糊集 A、B、C，空集用∅表示，模糊集的并、交、补运算具有以下的性质：

- 幂等律：$A\cup A=A$　　$A\cap A=A$
- 交换律：$A\cup B=B\cup A$　　$A\cap B=B\cap A$
- 结合律：$(A\cup B)\cup C=A\cup(B\cup C)$　　$(A\cap B)\cap C=A\cap(B\cap C)$
- 分配律：$A\cup(B\cup C)=(A\cup B)\cap(A\cup C)$　　$A\cap(B\cap C)=(A\cap B)\cup(A\cap C)$
- 吸收律：$(A\cap B)\cup A=A$　　$(A\cup B)\cap A=A$
- 同一律：$A\cup U=U$　　$A\cap U=A$　$A\cup\varnothing=A$　　$A\cap\varnothing=\varnothing$
- 复原律：$\bar{\bar{A}}=A$
- 对偶律：$\overline{A\cup B}=\bar{A}\cap\bar{B}$　$\overline{A\cap B}=\bar{A}\cup\bar{B}$

11.1.3 λ截集

模糊集的范围是不能明确画出的。但在实际应用中，往往需要对模糊现象做出明确的判定。因此，需要建立模糊集与经典集合的关系。截集概念即描述了这种转换关系。

设 $\underline{A}\in F(U)$，对任意 $\lambda\in[0,1]$，集合

$$A_\lambda = \{u \mid u\in U, \mu_{\underline{A}}(u) \geqslant \lambda\}$$

称为集合 $\underline{A}$ 的 λ 截集，λ 称为阈值或置信水平。

由定义可知，$\underline{A}$ 集合为模糊集，A_λ 为普通集，通过阈值实现了模糊集到普通集的转换。例如表 11.1 就为在不同阈值情况下，模糊集与截集间的关系。

表 11.1　模糊集与截集的关系

编　号	年　龄	$\underline{A}$(u)	$A_{0.6098}$(u)	$A_{0.22}$(u)
S_1	20	1	1	1
S_2	27	0.8621	1	1
S_3	29	0.6098	1	1
S_4	35	0.2000	0	1
S_5	40	0.1000	0	0

11.2　模糊关系

一般情况下，对于有限论域 $U=\{u_1,u_2,\cdots,u_n\}$，$V=\{v_1,v_2,\cdots,v_n\}$，称 $U\times V$ 上的模糊集 R 为从 U 到 V 的一个模糊关系，即对 $\forall(x,y)\in U\times V$，都指定它对 R 的隶属度 $\mu_R(u,v)$为：$\mu_R: U\times V\to[0,1]$。

U 与 V 之间的模糊关系还可用 n 行 m 列的模糊矩阵 $\underline{\boldsymbol{R}}$ 表示

$$\underline{R} = (r_{ij})_{n\times m}$$

其中：$r_{ij}=\mu_{\underline{R}}(u_i,v_j)$。

设 $\boldsymbol{R}$，$\boldsymbol{S}$ 皆为 m 行 n 列的模糊矩阵，则可通过模糊矩阵表示 $\underline{R}$ 与 $\underline{\boldsymbol{S}}$ 的并、交、补及 λ 截集：

$$R\cup S = (r_{ij}\vee s_{ij})$$

$$R\cap S = (r_{ij}\wedge s_{ij})$$

$$\overline{R} = (1-r_{ij})$$

$$R_\lambda = (\lambda r_{ij}) \quad \lambda r_{ij} = \begin{cases} 1 & r_{ij}\geqslant\lambda \\ 0 & r_{ij}\leqslant\lambda \end{cases}$$

根据模糊关系的定义，可以得到模糊关系的合成运算，即由 $\underline{Q}$ 和 $\underline{R}$ 构成的新的模糊关系 $\underline{Q}\circ\underline{R}$ 称为合成模糊关系

$$\mu_{\underline{Q}\circ\underline{R}}(u,w) = \bigvee_{v\in V}(\mu_{\underline{Q}}(u,v)\wedge\mu_{\underline{R}}(v,m))$$

当 U、V、W 均为有限论域时，即 $U=\{u_1,u_2,\cdots,u_n\}$,$V=\{v_1,v_2,\cdots,v_n\}$,$W=\{w_1,w_2,\cdots,w_n\}$，$\underline{Q}$，$\underline{R}$ 和 $\underline{S}=\underline{Q}\circ\underline{R}$ 均可表示为矩阵形式

$$\underline{Q} = (q_{ij})_{n\times m}, \underline{R} = (r_{jk})_{m\times l}, \underline{S} = s_{ik})_{n\times l}$$

其中：$s_{ik} = \bigvee_{j=1}^{m}(q_{ij}\wedge r_{jk})$。

如果 $\underline{R}$ 满足以下条件，则称 $\underline{R}$ 为论域 U 上的一个模糊等价关系。

（1）自反性，即 $\underline{R} \subset I$

（2）对称性，即 $\underline{R}^T = \underline{R}$

（3）传递性，即 $\underline{R} \circ \underline{R} \subset \underline{R}$

如果 $\underline{R}$ 满足以下条件，则称 $\underline{R}$ 为 U 上的模糊相似关系。

（1）自发性，即 $\underline{R} \subset I$

（2）对称性，即 $\underline{R}^T = \underline{R}$

从以上的定义可看出，为了从模糊相似关系得到模糊等价关系，可将模糊相似矩阵自乘，即 $\underline{R} \circ \underline{R} \underline{\Delta} \underline{R}^2, \underline{R}^2 \circ \underline{R}^2 \underline{\Delta} \underline{R}^4$，直到 $R^{2k} = R^k$。至此，$\underline{R}^k$ 便是模糊等价矩阵，它所对应的模糊关系便为模糊等价关系。

介绍等价关系的目的是为了将集合划分若干等价类。

设 $\underline{R}$ 是论域 U 上的等价关系，λ 从 1 下降到 0，依次截得等价关系 R_λ，它们都将 U 做了分类。由于满足条件

$$\lambda_2 \leqslant \lambda_1 \Rightarrow \mathrm{R}_{\lambda_2} \supset \mathrm{R}_{\lambda_1}$$

因此，$\forall u, v \in U$，若 u 与 v 相对于 $R_{\lambda1}$ 来说是属于同一类，$(u,v) \in R_{\lambda_1}$，则 $(u,v) \in R_{\lambda_2}$，即 u 与 v 相对于 $R_{\lambda2}$ 来说也属于同一类，这意味着由 $R_{\lambda2}$ 所得到的分类是由 $R_{\lambda1}$ 所得到的分类的加粗。

当 λ 从 1 下降到 0 时，分类由细变粗，逐渐归并，形成一个分级聚类树。

模糊关系主要用于模糊模式识别。模糊模式识别大致有两种方法：一种是直接方法，按“最大隶属原则”进行归类；另一种是阈值原则。

1. 最大隶属度原则

直接由计算样本的隶属度来判断其归属的方法，即为模式识别的最大隶属度原则。这种分类方式的效果十分依赖于建立已知模式类隶属函数的技巧。

设 $\underline{A}_1, \underline{A}_2, \cdots, \underline{A}_m \in F$（U），$x$ 是 U 中的一个元素

若 $\mu_{\underline{A}_i}(x) > \mu_{\underline{A}_j}(x)$（$j$=1,2,$\cdots$,$m$，$i \neq j$）

则 x 隶属于 $\underline{A}_i$，即将 x 判属于第 i 类。

2. 阈值原则

设 $\underline{A}_1, \underline{A}_2, \cdots, \underline{A}_m \in F$（$U$），$x$ 是 U 中的一个元素。取定水平 $\alpha \in [0,1]$，若 $\bigvee_{i=1}^{m} \mu_{\underline{A}_i}(x) < \alpha$，则不能识别；若存在 $i_1, i_2, \cdots, i_k$，使 $\mu_{\underline{A}_{i_k}}(x) \geqslant \alpha$，$j = 1,2,\cdots,k$，则 x 隶属于 $\bigcap_{j=1}^{k} \underline{A}_{i_j}$。

11.3 模糊聚类

模糊聚类分析是指一定的要求和规律将事物进行分类的一种数学方法，由于现实的分类往往伴随着模糊性，所以用模糊理论进行聚类会显得更自然，更符合客观实际。

模糊聚类方法有多种，例如传递闭包法、最大树法、编网法、模糊 K-均值方法等。它们能从原始数据中提取数据，对特征进行优化选择和降维，已广泛应用于经济学、生物学、气象学、信息科学等许多领域。

设 $X=\{x_1,x_2,\cdots,x_n\}$为被分类对象的全体，每一对象 x_i 由一组数据$\{x_{i1},x_{i2},\cdots,x_{im}\}$表征，即有 m 个分量（即 m 个特征）。对象之间的关系即为特征向量之间的关系，也即为 R^m 空间中元素之间的关系，聚类分析的基本思想是把 R^m 空间中具有某种特殊关系的对象聚合成一类。

11.3.1　数据标准化

在计算距离与相关系数时，一般需先对对象的特征量进行数据标准化，以消去量纲对分类的影响。常用的方法如下。

1. 标准差标准化

$$x_{ij}^{'}=\frac{x_{ij}-\mu_j}{s_j}, i=1,2,\cdots,n, j=1,2,\cdots,m$$

其中：μ_j和 s_j分别为 x_{ij}的均值和标准差。

2. 极差标准化

$$x_{ij}^{'}=\frac{x_{ij}-\mu_j}{S_j}, i=1,2,\cdots,n, j=1,2,\cdots,m$$

其中：S_j为极差，$S_j=\max\limits_{1\leqslant i\leqslant n}\{x_{ij}\}-\min\limits_{1\leqslant i\leqslant n}\{x_{ij}\}$。

11.3.2　相似系数和距离

在聚类分析中，最重要的是定义聚类关系，常见的是描述 R^m 空间中点与点关系的量，有距离与相似系数。如果将对象 $X=\{x_1,x_2,\cdots,x_n\}$看成是 R^m 空间中的 n 个点，从几何的角度看，可以定义这些点之间的距离 $d(x_i,x_j)$与相似系数 r_{ij}。

1. 数量积法

$$r_{ij}=\begin{cases}1, i=j\\ \dfrac{1}{M}\sum\limits_{k=1}^{m}x_{ik}x_{jk}, i\neq j\end{cases}$$

其中：$M=\max\limits_{i\neq j}\sum\limits_{k=1}^{m}x_{ik}x_{jk}$

显然 $r_{ij}\in[0,1]$，如果 r_{ij} 出现负值，可以用下面的方法将其调整为非负值。

①令 $r_{ij}^{'}=\dfrac{r_{ij}+1}{2}$，则 $r_{ij}^{'}\in[0,1]$；

②$r_{ij}^{'}=\dfrac{r_{ij}-\overline{m}}{\overline{M}-\overline{m}}(i\neq j)$，其中：$\overline{m}=\min\limits_{i\neq j} r_{ij}, \overline{M}=\max\limits_{i\neq j} r_{ij}$，则 $r_{ij}^{'}\in[0,1]$。

2. 夹角余弦法

$$r_{ij}=\frac{\sum_{k=1}^{m}x_{ik}x_{jk}}{\sqrt{\sum_{k=1}^{m}x_{ik}^{2}}\sqrt{\sum_{k=1}^{m}x_{jk}^{2}}}$$

如果 r_{ij} 出现负值，同样要将其调整为非负值。

3. 相关系数法

$$r_{ij}=\frac{\sum_{k=1}^{m}(x_{ik}-\overline{x}_{i})(x_{jk}-\overline{x}_{j})}{\sqrt{\sum_{k=1}^{m}(x_{ik}-\overline{x}_{i})^{2}}\sqrt{\sum_{k=1}^{m}(x_{jk}-\overline{x}_{j})^{2}}}$$

其中：$\overline{x_i}=\sum_{k=1}^{m}x_{ik}$，$\overline{x_j}=\sum_{k=1}^{m}x_{jk}$。

4. 最大最小法

$$r_{ij}=\frac{\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\sum_{k=1}^{m}(x_{ik}\vee x_{jk})}$$

5. 算术平均法

$$r_{ij}=\frac{2\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\sum_{k=1}^{m}(x_{ik}+x_{jk})}$$

6. 当 $x_{ik}, x_{jk}\geqslant 0$，可以采用几何平均最小法

$$r_{ij}=\frac{\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\sum_{k=1}^{m}\sqrt{x_{ik}x_{jk}}}$$

7. 绝对值指数法

$$r_{ij}=\mathrm{e}^{-\sum\limits_{k=1}^{m}|x_{ik}-x_{jk}|}$$

8. 指数相似系数法

$$r_{ij}=\frac{1}{m}\sum_{k=1}^{m}\mathrm{e}^{-(\frac{x_{ik}-x_{jk}}{s_k})^2}$$

需要适当选择 s_k。

9. 绝对值倒数法

$$r_{ij}=\begin{cases}1, i\neq j\\ \dfrac{\widehat{M}}{\sum\limits_{k=1}^{m}|x_{ik}-x_{jk}|}, i\neq j\end{cases}$$

需要适当 $\widehat{M}$ 选择，使 r_{ij} 在[0,1]中且分散。

10. 绝对值减数法

$$r_{ij}=1-c\sum_{k=1}^{m}|x_{ik}-x_{jk}|$$

需要适当 c，使 r_{ij} 在[0,1]中且分散。

11. 贴近法

如果特征 x_{ik}，$x_{jk}\in[0,1]$（$k=1,2,\cdots,m$），则 x_{ik}、x_{jk} 可以看作模糊向量，并以它们的贴近度为其相似程度。

贴近度是用来衡量两个模糊集 $\underline{A}$ 和 $\underline{B}$ 的接近程度，用 N（$\underline{A}$，$\underline{B}$）表示。贴近度越大，表明这两者越接近。常用的贴近度有以下三种：

（1）格贴近度。

$$r_{ij}=\begin{cases}1, & i=j\\ N(x_i,x_j)=(\bigvee\limits_{k=1}^{m}(x_{ik}\wedge x_{jk})\wedge(1-\bigwedge\limits_{k=1}^{m}(x_{ik}\wedge x_{jk}), & i\neq j\end{cases}$$

（2）距离贴近度。

$$r_{ij}=1-cd^{\alpha}(x_i,x_j)$$

其中：c、α 为适当选择参数值，$d(x_i,x_j)$为模糊集各种距离。

- 闵可夫斯基距离

$$d(x_i,x_j)(p)=(\sum_{k=1}^{m}|x_{ik}-x_{jk}|^p)^{\frac{1}{p}}$$

当 p=1 时，为海明距离

$$d(x_i,x_j)(1)=\sum_{k=1}^{m}|x_{ik}-x_{jk}|$$

当 p=2 时，为 Euclidean 距离

$$d(x_i,x_j)(2)=(\sum_{k=1}^{m}|x_{ik}-x_{jk}|^2)^{\frac{1}{2}}$$

- 切比雪夫距离

$$d(x_i,x_j)=\bigvee_{k=1}^{m}|x_{ik}-x_{jk}|$$

（3）其他贴近度。

$$n_1(x_i,x_j)=\frac{\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\sum_{k=1}^{m}(x_{ik}\vee x_{jk})}$$

$$n_2(x_i,x_j)=\frac{2\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\sum_{k=1}^{m}x_{ik}+\sum_{k=1}^{m}x_{jk}}$$

11.3.3 模糊聚类分析

模糊聚类分析可分为如下三步。

1. 建立模糊相似矩阵

建立模糊相似矩阵是实现模糊聚类的关键。设 $S=\{X^1,X^2,\cdots,X^N\}$是待聚类的全部样本，每一个样本都由 n 个特征表示

$$X^i=(x^i_1, x^i_2, \cdots, x^i_n)$$

第一步是求样本集中任意两个样本 X_i 与 X_j 之间的相似系数 r_{ij}，进而构造模糊相似矩阵 $\underline{R}=(r_{ij})_{N\times N}$。求相似系数的方法很多，可以根据需要选择其中的一种。

2. 将模糊相似关系变换为模糊等价关系

由第一步建立的模糊矩阵，一般情况下是模糊相似矩阵，即只满足对称性和自反性，不满足传递性，还需要将其改造成模糊等价矩阵。

3. 模糊聚类

对求得的模糊等价矩阵求 λ-截集，就可求得在一定条件下的分类情况。

11.3.4　模糊 K-均值聚类

模糊聚类算法常用的方法是模糊 K-均值算法（Fuzzy K-means，FKM），该算法是在传统 K-均值算法中应用了模糊技术。

FKM 算法把 n 个向量 x_i，$i=1,2,\cdots,n$，分成 K 个模糊集，并求得每个簇的聚类中心，使下述的目标函数达到最小

$$J_m(u,v)=\sum_{k=1}^{n}\sum_{i=1}^{K}u_{ik}^{m}d(x_k,v_i)$$

其中：$u_{ik}\in[0,1],\forall i,k;\sum_{i=1}^{K}u_{ik}=1,\forall k;0<\sum_{i=1}^{K}u_{ik}<n,\forall i;d(x_k,v_i)=\|x_k-v_i\|^2$，$m$ 为模糊权重指数，$1<m<+\infty$，聚类中心 v_i 和隶属度 u_{ik} 的计算如下：

$$v_i=\frac{\sum_{k=1}^{n}u_{ik}^{m}x_k}{\sum_{k=1}^{n}u_{ik}^{m}},i=1,2,\cdots,K$$

$$u_{ik}=\frac{1}{\sum_{j=1}^{K}(\frac{d_{ik}}{d_{jk}})^{\frac{1}{m-1}}},i=1,2,\cdots,K,k=1,2,\cdots,n$$

KFM 算法计算简单而且运算速度快，具有比较直观的几何意义，但与算法一样，只用类中心表示类的方法只适用发现球状类型的簇，在很多情况下，算法对噪声数据敏感。

11.4　基于 MATLAB 的模糊集处理技术

在 MATLAB 中，可以利用 Fuzzy Logic Toolbox 工具箱处理有关模糊集理论的问题。

例 3.1　在模糊系统的应用研究中，构建隶属度函数是一个比较关键的步骤。隶属度函数的构建可以有多种方法，如例化法、统计法、样板法等。但必须指出，迄今还没有一个一般的、普遍的法则，其构建多少还带有主观性和经验性的成分。

在环境保护中，环境质量评价是一个重要的方面，它是一个模糊评判过程。试构建各级标准水的隶属度函数。

解：

根据 GB 3838—1988《地面水环境质量标准》，一般将水质污染程度分为五类，即

```
B=[I, II, III, IV, V]
```

其中 I、II、III、IV 和 V 分级标准采用《地面水环境质量标准》中的值，据此，可构成以下的隶属度函数（以三级水中的 NH_3-N 为例）。首先编写一个隶属度函数表达式的 M 文件 NH3_Nmf：

```
function y=NH_3mf(x,params)
a =params(1); b =params(2); c =params(3);
%a 为二级标准值,b 为三级标准值,c 为四级标准值
y = zeros(size(x));
index = find(x==b);            % 当测量值等于此级标准值时,y=1
y(index) =ones(size(index));
index = find(a > =x | x >= c); % 当测量值小于二级标准或大于四级标准时,y=0
y(index) = zeros(size(index));
index = find(a< x & x < b);    %当测量值小于三级标准、大于二级标准时,y=(x-a)/(b-a)
y(index) = (x(index)-a)/(b-a);
index = find(x > b&x<c);       % 当测量值大于三级标准、小于四级标准时,y=(x-c)/(b-c)
y(index) = (x(index)-c)/(b-c);
```

在 MATLAB 工作空间输入以下命令：

```
>>x=0:0.01:3;water=newfis('water');water=addvar(water,'input','three_water',[0 3]);
>>mfedit(water);               %编辑隶属度函数
```

打开隶属度函数编辑器，选择 File 中的 import，选中 From Workspace…，打开 load FIS from workspace 对话框，在其 Workspace variable 栏中输入 water，单击 OK 关闭对话框，并打开 FIS Edit 对话框，在其 Edit 菜单中选择 Membership Functions，打开 Membership Functions Edit 对话框，选择其 Edit 菜单中的 Add Custom MF…，打开 Custom Membership Function 对话框，在 MF name 栏中输入隶属函数名字（在此为 a1），在 M-line function name 栏中输入 NH_3mf，在 parameter list 栏中输入[0.5 1 1.5]（此 3 个参数为各级标准临界值），单击 OK，便可以得到图 11.1 所示的三级标准中“硝酸盐－N”的隶属度函数，对于其他指标，用类似的方法同样可以得到其隶属度函数。

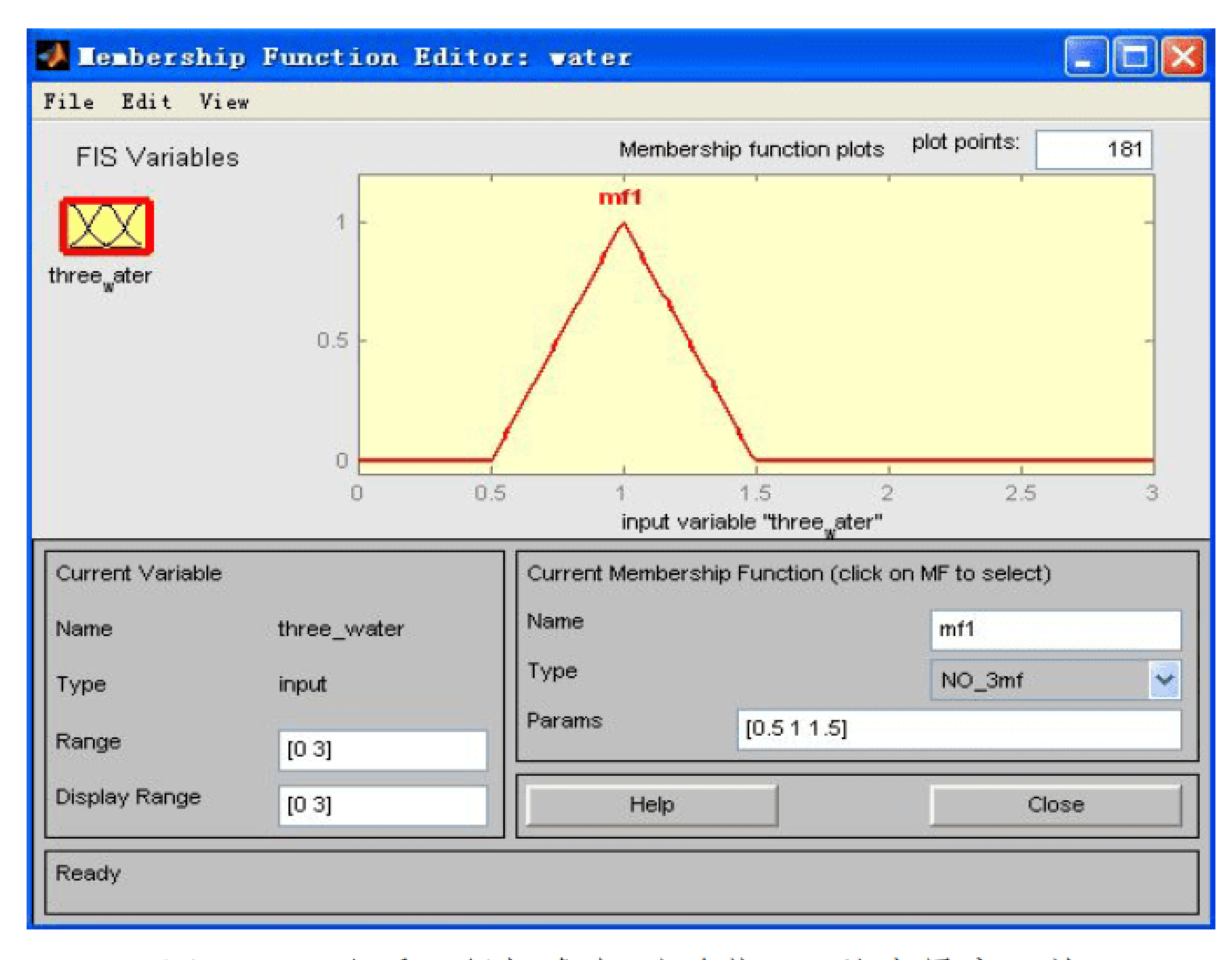


图 11.1　水质三级标准中“硝酸盐－N”的隶属度函数

例 3.2　为了研究气温与降水的关系，定义了如下的 4 个模糊集：

A_1：2 月份最低气温（≤5℃）的天数；

A_2：冬季极端最低气温；

A_3：极端最低气温出现时间；

A_4：冬季平均气温。

假设 4 个模糊集的隶属度函数分别如下：

$$f_{A_1}(x)=\begin{cases}0, & x\leqslant 4\\ \left[1+\left(\dfrac{x-4}{2}\right)^{-2}\right]^{-1}, & x>4\end{cases}$$

$$f_{A_2}(x)=\begin{cases}0, & x\leqslant -12\\ \left[1+\left(\dfrac{x+12}{2}\right)^{-2}\right]^{-1}, & x>-12\end{cases}$$

$$f_{A_3}(x)=\begin{cases}0, & x\leqslant 30\\ \dfrac{x-30}{30}, & 30<x<60\\ 1, & x\geqslant 60\end{cases}$$

$$f_{A_4}(x)=\begin{cases}1, & x\leqslant 3\\ 1-\dfrac{x-3}{0.6}, & 3<x<3.6\\ 1, & x\geqslant 3.6\end{cases}$$

根据以上这 4 个隶属度函数，构建表示冬季低温时间“长”、冬季气温“低”和冬季“冻大”程度的模糊集隶属度函数。

解：

根据题中给定的 4 个模糊集，可求出表示冬季低温时间“长”的模糊集

$$B=A_1\cup A_3$$

相应的隶属度函数

$$f_{A1\cup A3}(x)=\max | \mu_{A1}(x),\mu_{A3}(x) |$$

表示冬季气温“低”的模糊集

$$C=A_2\cup A_4$$

相应的隶属度函数

$$f_{A2\cup A4}(x)=\max | \mu_{A1}(x),\mu_{A4}(x) |$$

表示冬季“冻大”程度的模糊集

$$E=B\cap C$$

相应的隶属度函数

$$f_{B\cap C}(x)=\min\{\mu_B(x),\mu_C(x)\}$$

据此，可画出图 11.3 所示的这些模糊集的隶属度函数。

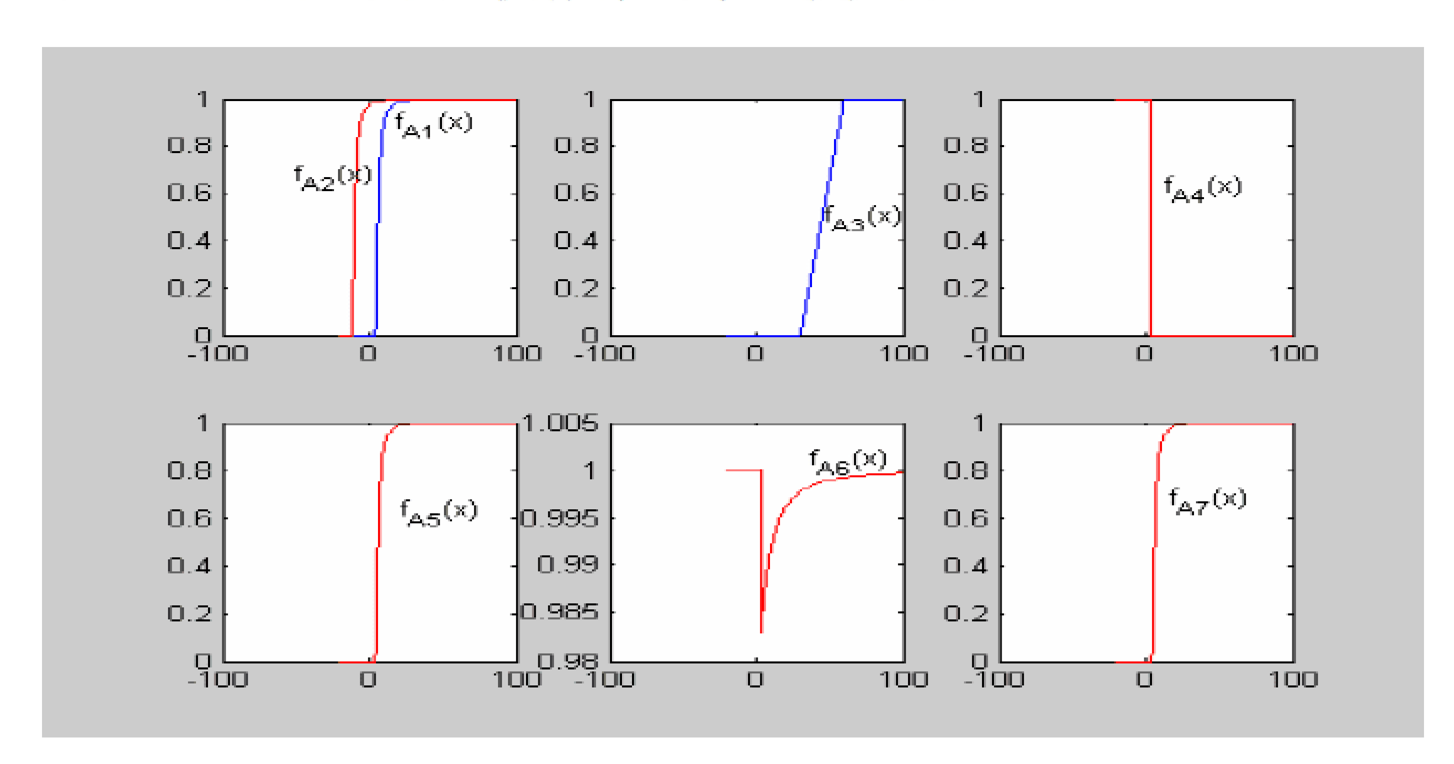


图 11. 3　各模糊集的隶属度函数

从此例可看出，可以利用模糊运算规则，进行“修饰词”的运算而得到“很大”“很长”等模糊集的隶属度函数。例如设隶属度函数为$f(x)$，则：

很高的隶属度函数为：$f^2(x)$

有点高的隶属度函数为：$f^{1/2}(x)$

低的隶属度函数为：$1-f(x)$

很低的隶属度函数为：$(1-f(x))^2$

有点低的隶属度函数为：$(1-f(x))^{1/2}$

中等的隶属度函数为：$\min\{f(x),\ 1-f(x)\}$

例3.3　模糊聚类可用于目标识别，在MATLAB中，模糊c-均值聚类函数为FCM。

设以下为新疆10个地区的集合：$X=[1,2,3,\ldots]$，其中1为阿勒泰、2为塔城、3为伊宁、4为昌吉、5为奇台、6为阿克苏、7为库车、8为喀什、9为和田、10为吐鲁番。

根据专业知识和实践经验，选取以下影响玉米生长的主要因素：

x_1：≥10℃积温（即一年中不小于 10℃的日平均温度累积）;

x_2：无霜期;

x_3：6～8 月平均气温;

x_4：5～9 月降水量

这些因子的实际观测值如表 11.3 所示。

表 11.3　玉米生长的主要影响因素

因　子 / 元　素	x_1	x_2	x_3	x_4
1	2704.7	149	21.3	83.1
2	2886.2	146	20.9	119.0
3	3412.1	175	21.8	139.2
4	3400.2	169	23.3	98.0
5	3096.4	157	22.3	105.0

续表

因子 / 元素	x_1	x_2	x_3	x_4
6	3798.2	207	22.6	42.4
7	4283.6	227	25.3	31.2
8	4256.3	222	24.5	40.7
9	4348.8	230	24.5	20.0
10	5378.3	221	31.4	8.3

请对后 9 个玉米种植地区进行分类，并求第 1 地区属于哪一类。

解：

```
>>load mydata.dat;y1=mean(x);y2=std(x);
>>x=[(x(:,1)-y1(1))/y2(1)  (x(:,2)-y1(2))/y2(2)  (x(:,3)-y1(3))/y2(3)  (x(:,4)-y1(4))/y2(4)];
%对数据归一化处理
>>[center,U,obj_fcn]=fcm(x(2:9,:),3);      %按南疆、北疆及吐鲁番三个地区
>>maxU=max(U);
>>index1=find(U(1,:)==maxU);index2=find(U(2,:)==maxU);index3=find(U(3,:)==maxU);
>>x(index1,1);                             %第 6、7、8、9 个数据即南疆地区为一类
>>x(index2,1);                             %显示为第 10 个数据即吐鲁番单独为一类
>>x(index3,1);                             %第 2、3、4、5 个数据即北疆地区为一类
>>x1=[ -1.2823 -1.2026 -0.8170 0.3154];    %新目标值
>>e=ones(3,1)*x1;                          %使新数据维数与分类值相等
>>f=(center-e)';                           %新数据与各聚类中心的值
>>ff=sum(f.^2);                            %最小二乘法
>>[min1,index]=min(ff);
>>disp(['新目标为第',num2str(index),'类'])
```

新目标为第 3 类，即归纳于北疆地区。

例3.4　对例3.3的数据采用模糊减法聚类。

解：

```
>>load mydata.dat;y1=mean(x);y2=std(x);
>>x=[(x(:,1)-y1(1))/y2(1)  (x(:,2)-y1(2))/y2(2)  (x(:,3)-y1(3))/y2(3)  (x(:,4)-y1(4))/y2(4)];
 >>figure,hold on;>>plot(x(:,1),x(:,2),'+')       %用二维图近似表示分类情况
 >>radii=0.3;                                     %半径值
 >>[c,s]=subclust(x,radii);                       %圆圈处代表聚类中心
 >>radii=0.5;[c,s]=subclust(x,radii);             %五角星处代表聚类中心
 >>plot(c(:,1),c(:,2),'kpentagram','markersize',15,'LineWidth',1.5)
```

从图 11.4 可看出，当半径为 0.3 时得到 8 个聚类中心，而为 0.5 时，只得到了 4 个聚类中心。

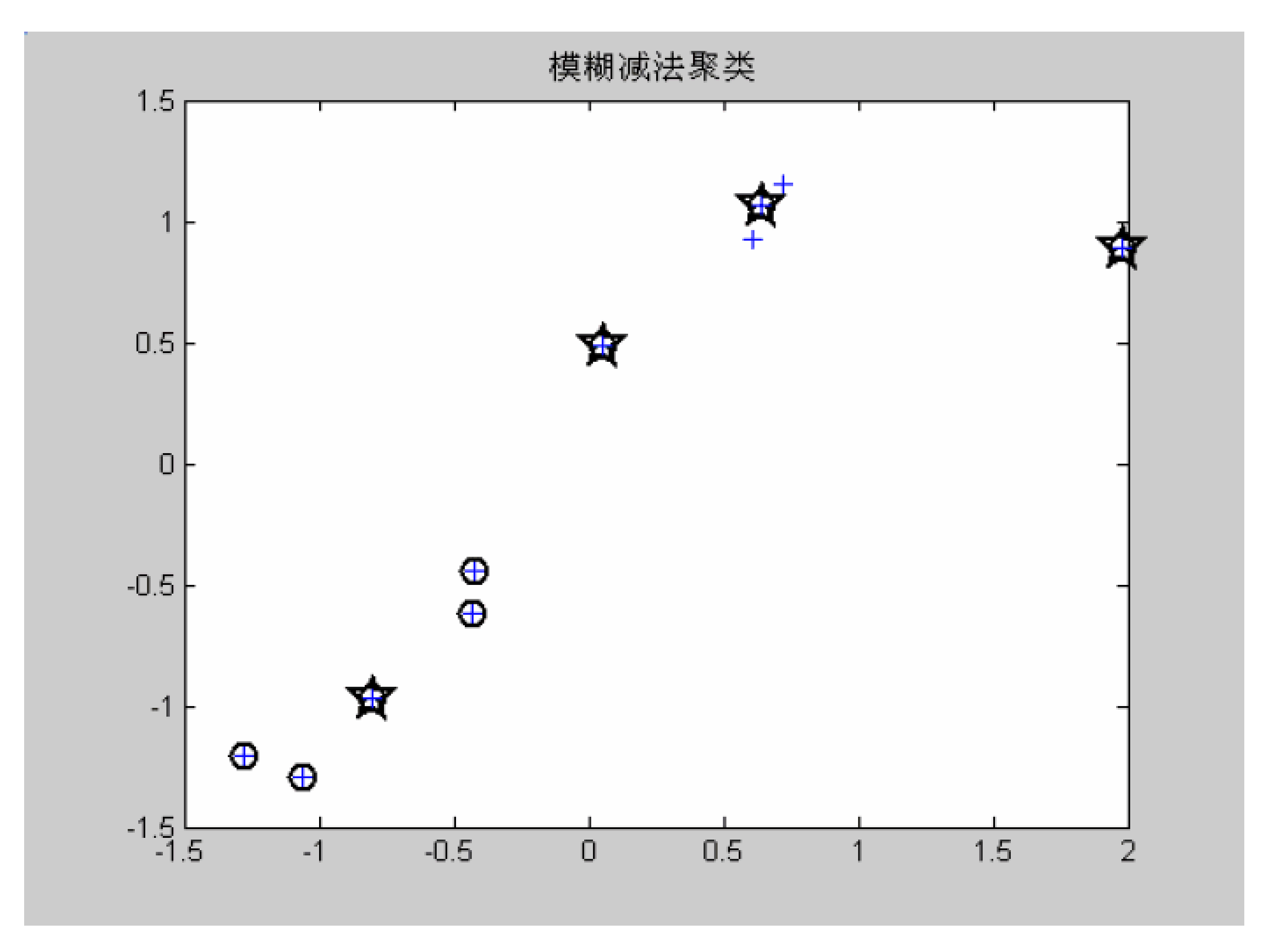


图 11.4　模糊减法聚类结果（radii=0.3 and 0.5）

例 3.5　某地区 1993—2000 年 1 月份水环境测量值及其相应的标准值如表 11.4 和表 11.5 所示，试对该地区该时期的水环境质量进行模糊评价。

表 11.4　各级环境标准指标值

序　号	指　标	I 类	II 类	III 类	IV 类	V 类
1	溶解氧	≥7.5	6	5	3	2
2	高锰酸钾指数	≤2	4	6	10	15
3	BOD_5	≤3	3	4	6	10
4	NH_3-N	≤0.15	0.5	1	1.5	2
5	挥发酚	≤0.002	0.002	0.005	0.01	0.1

表 11.5　该地区水环境各指标的测量值

序　号	溶解氧	高锰酸钾指数	BOD_5	NH_3-N	挥 发 酚
1993	10.2	1.8	3.5	1.16	0.0
1994	9.2	1.9	7.1	2.33	0.004
1995	8.0	4.1	7.6	0.23	0.004
1996	10.3	1.6	1.5	0.34	0.0
1997	5.8	8.2	6.6	3.91	0.031
1998	3.2	9.4	12.8	6.88	0.00
1999	7.7	4.0	6.6	0.99	0.006
2000	7.4	4.6	7.1	3.67	0.0

解：

首先根据表 11.4 确定各级的隶属度函数。以溶解氧为例，其一、二级隶属度为：

$$f_I(x)=\begin{cases}0 & x\leqslant 6\\ \dfrac{x-6}{7.5-6} & 6<x<7.5\\ 1 & x\geqslant 7.5\end{cases}\qquad f_{\rm II}(x)=\begin{cases}0 & \leqslant 5\text{或}x\geqslant 7.5\\ \dfrac{7.5-x}{7.5-6} & 6\leqslant x<7.5\\ \dfrac{x-5}{6-5} & 5<x<6\end{cases}$$

采用类似的方法建立其余各级的隶属度的函数。

然后对每一个指标值进行单因素评价，得到综合评判矩阵 R。

再确定因素重要程度模糊集即每项指标的权重。权重可以用下式计算

$$W_i=(c_i/S_i)/\left(\sum\frac{c_i}{S_i}\right)$$

其中：c_i是测定值，S_i是某项指标的各分级指标的平均值。需要注意的是，因为溶解氧的值越大越好，所以权重取 W_i的倒数。得到各指标重要程度的模糊集 A。

最后选用模型 M（∧，∨），得模糊评价集：B = A*R

根据计算结果便可以知道评价结果。

```
>>load x; m=size(x,1);
>> for k=1:m;x1=x(k,:);y=water_mu(x1);y1(k,:)=fuzzmu(y(:,end)',y(:,1:end-1));end
>> y1
 y1 =0.1841      0.2689       0.4499       0.3200            0       %三级
     0.1120      0.0368       0.0368       0.2992       0.4957       %五级
     0.1962      0.1850       0.0561       0.4881       0.4000       %四级
     0.3534      0.2557            0            0            0       %一级
          0      0.0978       0.1337       0.1572       0.4581       %五级
          0           0       0.1069       0.1236       0.5622       %五级
     0.1680      0.1488       0.2645       0.3493       0.1500       %四级
     0.1027      0.1005       0.1005       0.2207       0.5761       %五级
```

例 3.6　逼近未知的非性函数有许多方法，如多项式逼近、指数函数逼近、人工神经网络逼近等。以模糊逻辑系统为基础的模糊模型也可用于非线性动态的建模，并显示出优良的性能。利用模糊推理系统对非线性函数 $f(x)=2e^{-x}\sin(x)$进行逼近。

解：

设定输入 x 的范围为[0,10]，并将它模糊分割成 5 个区，即设定一个隶属度函数，其类型采用广义的钟形函数，则：

```
>>x=[0:0.1:10]';y=2*exp(-x).*sin(x);data=[x y];
>>mf_type='gbellmf';                                    %训练选项
>>mf_n=5;
>>fis1=genfis1(data,mf_n,mf_type);                      %产生 FIS 结构的初值
```

```
>>epoch=50; errorgoal=0; step=0.01;                    %训练参数
>>trnOpt=[epoch errorgoal step NaN NaN];disOpt=[1 1 1 1];chkData=[];
>>[fis2,error,st,fis3,e2]=anfis(data,fis1,trnOpt,disOpt,chkData);
>>xx=data(:,1);yy=evalfis(xx,fis2);                    %求模拟输出值
>>rmse=norm(yy-data(:,2))/sqrt(size(xx,1));            %求均方误差
```

图 11.5 为训练结果。

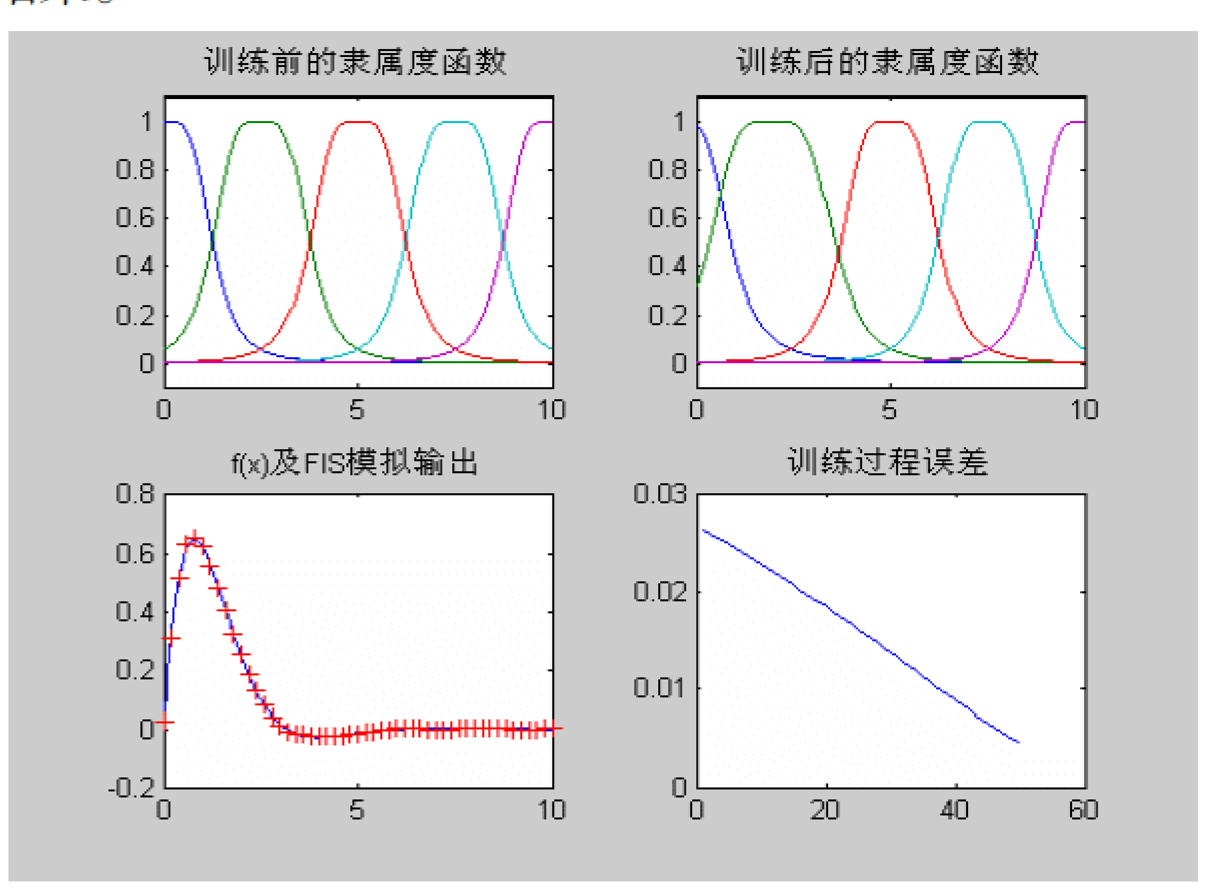


图 11.5　函数逼近的 ANFIS 训练结果

例 3.7　在利用 genfis1 逼近非线性系统时，当数据维数增加，很明显计算量将大大增多，此时可利用 genfis2 产生 FIS 初始结构。如有一个故障诊断系统，其故障编码为表 11.6 所示。请对此系统进行模拟逼近。

表 11.6　测试结果

故障序号	测试编码	故障编码
1	11111	00000
2	01000	10000
3	10000	01000
4	11000	00100
5	11100	00010
6	11110	00001

解：

用前 5 个数据进行训练，最后一个数据用于检验：

```
>>x_in=[1 1 1 1 1;0 1 0 0 0;1 0 0 0 0;1 1 0 0 0;1 1 1 0 0];
>> x_out=['00000';'10000';'01000';'00100';'00010'];x_out=bin2dec(x_out);data=[x_in x_out];
>>%anfis 格式只允许 1 列输出，将故障编码改为十进制
>>fismat=genfis2(x_in,x_out,0.5,minmax(data')');
>>epoch=50; errorgoal=0; step=0.01;          %训练参数
>>trnOpt=[epoch errorgoal step NaN NaN];disOpt=[1 1 1 1];chkData=[];
```

```
>>[fis2,error,st,fis3,e2]=anfis(data,fismat,trnOpt,disOpt,chkData);
>>x1=[1 1 1 1 0];yy=evalfis(x1,fis2);
 yy =1.0000
>>dec2bin('yy')                          %显示四位，前二位为补位
 ans =110001
即故障编码为 00001
```

例 3.8　考虑煤炭按成因分类的模糊识别问题。根据成因可将煤炭分为三大类，即无烟煤 A_1、烟煤 A_2 和褐煤 A_3。设论域 U 为所有煤种的集合，无烟煤 A_1、烟煤 A_2 和褐煤 A_3 是 U 上的模糊子集，对于某一给定的具体煤种 u，试判断其归属。所用数据如表 11.7 所示。

表 11.7　各个煤样特性指标的测量值

煤样样本分类及编号	序号	特性指标(u)									
		碳	氢	硫	氧	镜质组分	丝质组分	块状微粒体	粒状微粒体	壳质树脂体	平均最大反射率
无烟煤 (A_1)	1	92.21	2.74	0.84	3.58	86.70	13.30	0.00	0.00	0.00	4.92
	2	92.58	2.80	1.00	2.98	90.01	9.70	0.20	0.00	0.00	3.98
	3	92.63	3.04	0.74	2.64	89.10	10.60	0.30	0.00	0.00	4.12
	4	93.01	1.98	0.55	3.46	89.00	9.40	0.80	0.00	0.00	6.05
	5	93.01	2.79	0.79	2.67	88.30	11.70	0.00	0.00	0.00	4.50
平均值		92.68	2.67	0.78	3.06	88.62	10.94	0.26	0.00	0.00	4.71
烟煤 (A_2)	6	84.62	5.61	0.76	7.30	69.10	13.10	1.4 0	4.10	12.50	0.90
	7	84.53	5.55	0.70	7.36	64.60	8.10	3.00	11.3	11.00	0.85
	8	83.82	5.78	0.90	7.80	84.10	2.70	1.20	7.40	4.50	0.93
	9	82.65	5.57	2.48	7.19	77.20	9.10	2.70	3.20	7.80	0.83
	10	82.43	5.77	1.61	8.53	84.90	3.80	2.30	5.00	4.10	0.84
	11	81.88	5.87	2.94	7.39	80.30	4.30	3.30	7.80	4.30	0.71
平均值		83.32	5.69	1.56	7.59	76.70	6.85	2.31	6.46	7.36	0.84
褐煤 (A_3)	12	72.49	5.31	2.11	20.23	85.72	7.90	3.54	3.12	3.73	0.30
	13	72.29	5.26	1.02	20.43	85.60	4.60	3.30	2.80	3.70	0.31
	14	71.39	5.33	1.07	21.03	84.70	5.90	2.80	3.00	3.60	0.32
	15	70.95	5.04	1.50	21.10	81.85	7.25	2.75	2.94	3.21	0.33
	16	71.85	5.17	1.14	20.95	85.10	7.21	3.54	2.77	3.54	0.32
平均值		71.79	5.22	1.36	20.74	84.59	6.57	3.18	2.92	3.55	0.31

解：

在模糊模式识别中，构造模糊模式的模糊函数是其关键和难点。下面介绍常用的样板法。

（1）设 U 为待识别对象全体的集合，$A_1,A_2,\cdots,A_p$ 为 U 上 p 个模糊模式，每一个识别对象 $u\in U$ 的特性指标向量为 $u=(u_1,u_2,\cdots,u_m)$。

从模糊模式 A_i 中选出个 k_i 样板，设为

$$a_{ij}=(a_{ij1},a_{ij2},\cdots,a_{ijm})\quad(i=1,2,\cdots,p;j=1,2,\cdots,k_i)$$

式中：a_{ij}表示第 i 个模糊模式 A_i中的第 j 个样板的特性指标向量；a_{ijk}表示第 i 个模糊模式 A_i中的第 j 个样板的第 k 个特性指标的实测数据。

（2）计算模糊模式 A_i中的 k_i个特性指标向量 $a_{ij}(i=1,2,\cdots,p;\ j=1,2,\cdots,k_i)$的平均值 a_i，即

$$a_i=(a_{i1},a_{i2},\cdots,a_{im})$$

式中：$a_{ik}=\frac{1}{k_i}\sum_{j=1}^{k_i}a_{ij_k}\qquad k=1,2,\cdots,m$

称 a_i 为模糊模式 A_i的均值样板。

（3）计算模糊模式 A_i的隶属函数

计算识别对象 $u=(u_1,u_2,\cdots,u_m)$与均值样板 $a_i=(a_{i1},a_{i2},\cdots,a_{im})$之间的距离 $d_i(u,a_i)$，如取欧氏距离，有：$d_i(u,a_i)=(\sum_{j=1}^{m}(u_j-a_{ij})^2)^{1/2}\qquad i=1,2,\cdots,p$

令 $D=\max\{d_1(u,a_1),d_1(u,a_2)\ \ \cdots,d_p(u,a_p)\}$

则模糊模式的隶属函数为：$A_i(u)=1-\frac{d_i(u,a_i)}{D}\quad i=1,2,\cdots,p$

根据以上所述，计算各模式的隶属度值过程如下。

（1）用 mean 函数求平均值，即每类的中心。

（2）计算待识别煤样与均值的距离（即与每类的距离），可以采用各种距离，如用欧氏距离，则可以用 norm 函数计算。

$$d_1(u,a)=(\sum_{j=1}^{10}(u_j-\overline{a_j})^2)^{1/2},d_2(u,b)=(\sum_{j=1}^{10}(u_j-\overline{b_j})^2)^{1/2},d_3(u,c)=(\sum_{j=1}^{10}(u_j-\overline{c_j})^2)^{1/2}$$

令 $D=d_1(u,a)+d_2(u,b)+d_3(u,c)$，则可得到三种类型煤的隶属函数：

$$A_1(u)=1-\frac{d_1(u,a)}{D},A_2(u)=1-\frac{d_2(u,b)}{D},A_3(u)=1-\frac{d_3(u,c)}{D}$$

从而可计算出每个煤样隶属于每种类型煤的隶属度值，据此可判断出其隶属的类别，计算结果符合实际（前 5 种煤样的计算结果）。

```
>> load x;c=mean(x(12:end,:));b=mean(x(6:11,:));a=mean(x(1:5,:));
>> for k=1:5
     d1(k,1:3)=0;
       for i=1:10
     d1(k,1)=d1(k,1)+sqrt((x(k,i)-a(i))^2);d1(k,2)=d1(k,2)+sqrt((x(k,i)-
b(i))^2);
        d1(k,3)=d1(k,3)+sqrt((x(k,i)-c(i))^2);
       end
     end
>> D1=sum(d1')';    %
>>for i= 1:5;for j=1:3;A1(i,j)=1-d1(i,j)/D1(i);end;end
```

例 3.9　胃病病人和非胃病人的生化指标测量值如表 11.8 所示。试用模糊神经网络方法对某未知样进行判别。

表 11.8　胃病病人和非胃病人生化指标的测定值

胃病类型	铜蓝蛋白（x_1）	蓝色反应 (x_2)	吲哚乙酸 (x3)	中性硫化物 (x_4)	归　类
胃病	228	134	20	11	1
	245	134	10	40	1
	200	167	12	27	1
	170	150	7	8	1
	100	167	20	14	1
非胃病	225	125	7	14	2
	130	100	6	12	2
	150	117	7	6	2
	120	133	10	26	2
	160	100	5	10	2
	185	115	5	19	2
	170	125	6	4	2
	165	142	5	3	2
	185	108	2	12	2
未知样	100	117	7	2	

解：

模糊神经网络计算的步骤如下：

① 对于k维输入量$x=[x_1,x_2,\dots,x_k]$，首先根据模糊规则计算各输入变量x_j的隶属度，隶属度函数采用高斯型：

$$\mu_{A_j^i} = \exp(-(x_j - c_j^i)2 / b_j^i) \qquad j = 1,2,\cdots,k; i = 1,2,\cdots,n$$

其中：c_j^i, b_j^i 分别为隶属度函数的中心和宽度；k为输入参数的维数（即特征向量数）；n为模糊子集数。

② 将各隶属度进行模糊计算，模糊算子采用连乘算子

$$\omega^i = \mu_{A_j^1}(x_1) * \mu_{A_j^2}(x_2) * \cdots * \mu_{A_j^k}(x_k) \qquad i = 1,2,\cdots,n$$

③ 根据模糊计算结果计算模糊模型的输出值

$$y_i = \sum_{i=1}^{n} \omega^i (p_0^i + p_1^i x_1 + \cdots + p_k^i x_k) / \sum_{i=1}^{n} \omega^i$$

④ 计算误差

$$e = \frac{1}{2}(y_d - y_c)^2$$

⑤ 系数修正

$$p_j^i(k) = p_j^i(k-1) - \alpha \frac{\partial e}{\partial p_j^i}, \qquad \frac{\partial e}{\partial p_j^i} = (y_d - y_c)\omega^i / \sum_{i=1}^{n} \omega^i x_j$$

⑥ 参数修正

$$c_j^i(k)=c_j^i(k-1)-\beta\frac{\partial e}{\partial c_j^i},b_j^i(k)=b_j^i(k-1)-\beta\frac{\partial e}{\partial b_j^i}$$

本例中参数的修正方法采用遗传算法。由于输入数据为四维，输出数据为一维，所以模糊神经网络的结构设为 4－8－1，即有 8 个隶属度函数，选择 5×8 个系数 p_0~p_4，c_j^i,b_j^i 各为 8×4 的矩阵，所以共有 104 个待优化系数。

编写适应度函数如下：

```
function y=m1(x)
xdata=[228 134 20 11;245 134 10 40;200 167 12 27; 170 150 7 8;
   100 167 20 14;225  125 7 14;130 100 6 12;150 117 7 6;120 133 10 26;
160 100 5 10;185 115 5 19;170 125 6 4;165 142 5 3;185 108 2 12;100 117 7 2]';
xdata=guiyi(xdata);ydata=[1 1 1 1 1 2 2 2 2 2 2 2 2 2];
I=4;M=8;[n,m]=size(xdata);                              %计算待测样时从这一行开始
p0(1:8)=x(1:8);p1(1:8)=x(9:16);p2(1:8)=x(17:24);p3(1:8)=x(25:32);p4(1:8)=x(33
:40);
c=reshape(x(41:72),8,4);b=reshape(x(73:104),8,4);       %将x分配给各个参数
y=0;
for k=1:m-1                             %计算待测样时其中的x值都要为测试样归一化的值
  for i=1:I;for j=1:M;u(i,j)=exp(-(xdata(i,k)-c(j,i))^2/b(j,i));end;end   %参数
模糊化
  for i=1:M;w(i)=u(1,i)*u(2,i)*u(3,i)*u(4,i);end         %隶属度计算
  addw=sum(w);
  for i=1:M

yi(i)=p0(i)+p1(i)*xdata(1,k)+p2(i)*xdata(2,k)+p3(i)*xdata(3,k)+p4(i)*xdata(4,
k);  %输出
  end
  addyw=0;addyw=yi*w';yn(k)=addyw/addw;                 %预测值,计算待测样时到此结束
  y=y+(ydata(k)-yn(k))^2/2;
end
```

打开遗传算法工具箱GUI，并在相应的框中输入各参数就可以进行计算。其中边界约束：Lower输入0.01*ones(1,104)；Upper输入10*ones(1,104)，种群规模选50。

计算结束后，将结果输出到命令窗口，即可以得到各个参数。利用这些参数和适应度函数的程序就可以计算未知样的归属，结果为：y_n =1.9825，属于第二类。

例 3.10　某地 1985—1995 年期间每年 10 月份的地下水平均值如下所示，试对该地的地下水位情况进行预测。

年份：1985　1986　1987　1988　1989　1990　1991　1992　1993　1994　1995

水位：27.33　26.92　26.40　25.87　25.42　25.12　24.93　24.89　24.73　24.56　24.60

解：

对于时间序列预报，希望通过目前时刻t为止已知的序列值来预报将来$t+p$时刻的序列值。首先构筑一个输入矩阵，设延迟时间为3，也即利用时间序列的前3个值来预测第4个值；然后再利用模糊—神经网络进行预测，程序如下：

```
>> x=[27.33 26.92 26.40 25.87 25.42 25.12 24.93 24.89 24.73 24.56 24.60];
>> m=3;n=length(x); for i=m+1:n;for j=1:m;x1(i,j)=x(i-(m-j+1));end;end
>> x1=x1(m+1:end,:);y=x(m+1:end);yy=[x1 y'];        %输入向量，即训练数据
>> fis1=genfis1(yy(1:end,:),3);
>> epoch=150; errorgoal=0; step=0.01;trnOpt=[epoch errorgoal step NaN
NaN];disOpt=[1 1 1 1];
>> chkData=[];                                       %检验数据
>> [fis2,error,st,fis3,e2]=anfis(yy,fis1,trnOpt,disOpt,chkData);
>> pred=evalfis(yy(:,1:3),fis2);                     %对数列预测
ans=25.8700  25.4200  25.1200  24.9300  24.8900  24.7300  24.5600  24.6000
```

读书笔记

第 12 章

粗糙集技术

12.1 粗糙集理论的基本概念

在自然界中，大部分事物所呈现的信息都是不完整和模糊的。对于这些信息，经典逻辑由于无法准确地描述，所以也就不能正确地处理。长期以来许多逻辑学家和哲学家都致力于研究模糊概念。但在现实世界中，并不能简单地用好坏、真假等确切的概念表示许多含糊现象，特别是在于集合的边界上，也即存在一些个体，既不能说它属于某个子集，也不能说它不属于该子集。

1965 年，Zadeh 提出了模糊集概念，之后经过半个世纪的努力，已经形成了较为完整的模糊集理论。

20 世纪 80 年代，波兰的科学家 Z・Pawlak 提出了粗糙集（rough set）理论。粗糙集用上、下近似两个集合来逼近任意一个集合，该集合的边界区域被定义为上近似集和下近似集的差，边界区域就是那些无法归属的个体。上、下二近似集合可以通过等价关系给出确定的描述，边界域的元素数目可以被计算出来。

12.1.1 知识表达系统和决策表

知识是对某些客观对象的认识。为了处理数据，需要对知识进行符号表示。知识表示系统就是研究对象的知识通过指定对象的基本特征和特征值来描述，以便通过一定的方法从大量的数据中发现有用的知识或决策规则。

知识表达系统可用下式表示：

$$S=<\widetilde{X},\ C,\ D,\ V,\ f>$$

其中：$\widetilde{X}$ 为对象的集合，即为论域；$C\cup D=R$ 是属性的集合；子集 C 和 D 分别称为条件属性的结果属性；$V=\bigcup_{r\in R}V_r$ 是属性值的集合，V_r 表示了属性 $r\in R$ 的属性范围；f：$\widetilde{X}\times R\to V$ 是一个信息函数，它指定 $\widetilde{X}$ 中每一对象 x 的属性值。

知识表达系统的数据以关系表的形式表示，关系表的行对应要研究的对象，列对应对象的属性，对象的信息通过指定对象的各属性值来表示。

设 $S=(\widetilde{X},A)$ 为一知识表达系统，且 C，$D\subseteq A$ 是两个属性子集，分别称为条件属性和决策属性，具有条件属性和决策属性的知识表可表达为决策表，记为 $T=(\widetilde{X},A,C,D)$ 或简称为 CD 决策表。关系 ind（C）和 ind（D）的等价关系分别称为条件类和决策类。

对象的特征由条件属性描述，决策属性表示该对象的分类。决策属性可能表示专家根据条件属性描述所做的分类、采取的行动或决策。

12.1.2 等价关系

设 A 代表某种属性集合。a 代表属性中的某一种取值。如果有两个样品 $\boldsymbol{X}_i$、$\boldsymbol{X}_j$，满足以下关系：

对于 $\forall a\in A,A\subset R,X_i,X_j\in\widetilde{X}$，它们的属性值相同，即 $f_a(\boldsymbol{X}_i)=f_a(\boldsymbol{X}_j)$ 成立，称对象 $\boldsymbol{X}_i$ 和 $\boldsymbol{X}_j$ 是对属性 A 的等价关系，表示为

$$IND(A)=\{(X_i,X_j)\mid(X_i,X_j)\in\widetilde{X}\times\widetilde{X},\forall a\in A,f_a(X_i)=f_a(X_j)\}$$

即属性相同的两个样品之间的关系为等价关系。

粗糙集的等价概念与传统的集合论的等价概念有本质的区别，在传统集合论中，当两个集合有完全相同的元素时，它们是等价的；而在粗糙集中，只是在某一个属性之下，集合的取值相等，它是集合间的拓扑结构，不是构成集合的元素间的比较。

在 $\widetilde{X}$ 中，对属性集 A 中具有相同等价关系的元素集合成为等价关系 IND（A）的等价集$[X]_A$，表示在属性 A 下与 X 具有等价关系的元素集合。

$$[X]_A = \{X_j \mid (X, X_j) \in IND(A)\}$$

12.1.3 等价划分

从所采集的训练集中把属性值相同的样品聚类，形成若干个等价集，构成 A 集合。在 $\widetilde{X}$ 中对属性 A 的所有等价集形成的划分表示为

$$A = \{E_i \mid E_i = [X]_A, i = 1, 2, \cdots,\}$$

具有特性：

（1）$E_i \neq \varnothing$；

（2）当 $i \neq j \in$时，$E_i \cap E_j = \varnothing$；

（3）$\widetilde{X} = \bigcup E_i$。

12.1.4 上近似集和下近似集

属性 A 可划分为若干个等价集，与决策集 Y 对应关系分上近似集 A^-（Y）和下近似集 A_-（Y）两种：

1. 下近似定义

对任意一个决策属性的等价集 $Y(Y \subseteq \widetilde{X})$，属性 A 的等价集 $E_i=[X]_A$，有

$$A_-(Y) = \bigcup\{E_i \mid E_i \in A \wedge E_i \subseteq Y\}$$

或

$$A_-(Y) = \{X \mid [X]_A \in Y\}$$

表示等价集 $E_i=[X]_A$ 中的元素都属于 Y，即 $\forall X \in A_-(Y)$，则 X 一定属于 Y。A_-（Y）表示下近似集。

2. 上近似定义

对任意一个决策属性的等价集 Y（$Y \subseteq \widetilde{X}$），属性 A 的等价集 $E_i=[X]_A$，有

$$A^-(Y) = \bigcup\{E_i \mid E_i \in A \wedge E_i \cap Y \neq \varnothing\}$$

或

$$A^-(Y) = \{X \mid [X]_A \cap Y \neq \varnothing\}$$

表示等价集 $E_i=[X]_A$ 中的元素可能属于 Y，即 $\forall X \in A^-(Y)$，则 X 可能属于 Y，也可能不属于 Y。A^-（Y）表示上近似。

3. 正域、负域和边界的定义

全集 $\widetilde{X}$ 可以划分为 3 个不相交的区域，即正域（POS_A）、负域（NEG_A）和边界（BND_A）。

正域：$POS_A(Y) = A_-(Y)$

负域：$NEG_A(Y) = \widetilde{X} - A^-(Y)$

边界：$BND_A = A^-(Y) - A_-(Y)$

由此可见：

$$A^-(Y) = A_-(Y) + BND_A(Y)$$

从上述的定义中可知，任意一个元素 $X \in POS(Y)$，一定属于 Y；任意一个元素 $X \in NEG(Y)$，一定不属于 Y；集合的上近似是其正域和边界的并集，即

$$A^-(Y) = POS_A(Y) \cup BND_A(Y)$$

对于元素 $X \in BND(Y)$，无法确定其是属于 Y，因此对于任意元素 $X \in A^-(Y)$，只知道 X 可能属于 Y。

12.1.5 粗糙集

若 $A^-(Y) = A_-(Y)$，即 $BND_A(Y) = \varnothing$，即边界为空，称 Y 为 A 的可定义集，否则 Y 为 A 的不可定义集，即 $A^-(Y) \neq A_-(Y)$，称 Y 为 A 的粗糙集（Rough set）。

12.1.6 粗糙集的非确定性的精确度 $\alpha_A(Y)$和粗糙度 $\rho_A(Y)$

集合的不确定性是由于边界的存在而引起的，集合的边界域越大，其精确性越低。为了准确地表达这一点，常用精确度 $\alpha_A(Y)$和粗糙度 $\rho_A(Y)$来表示，即

$$\alpha_A(Y) = \frac{|\widetilde{X}| - |A^-(Y) - A_-(Y)|}{|\widetilde{X}|}$$

上式中$|\widetilde{X}|$和$|A^-(Y) - A_-(Y)|$分别为集合$[\widetilde{X}]$、$[A^-(Y) - A_-(Y)]$中的记录总数，精确度用来反映 $\widetilde{X}$ 的知识的完整程度，即能够根据 $\widetilde{X}$ 中各属性的属性值就能够确定其属于或不属于 Y 的比例。

也可以用粗糙度来定义集合 $\widetilde{X}$ 的不确定程度，即

$$\rho_A(Y) = 1 - \alpha_A(Y)$$

与概率论或模糊集合不同，不粗糙集的精确的数不是事先假定的，而是通过表达知识不精确性的概念近似计算的，这样不精确的数值表示有限知识的结果。

12.2 分类规则的形成

应用粗糙集理论，对数据进行学习，从中寻找隐含的模式和关系，对数据进行约简，评价数据的重要性，从数据中产生分类规则。

通过分析 $\widetilde{X}$ 中的两个划分 Y 和 X 之间的关系，把 Y 视为分类条件，X 视为分类结论，可得到下面的分类规则：

（1）当 $Y \cap X \neq \varnothing$，则有：des(Y)→des(X)

des(Y)和 des(X)分别是等价集 Y 和等价集 X 中的特征描述：

① 当 $Y \cap X = Y$，即 Y 全部被 X 包含，此时建立的规则是确定的，规则的置信水平为 1；

② 当 $Y \cap X \neq Y$，即 Y 全部不被 X 包含，此时建立的规则是不确定的，规则的置信水平为

$$cf = \frac{|Y \cap X|}{|Y|}$$

（2）当 $Y \cap X = \varnothing$，Y 和 X 不能建立规则。

12.3　知识的约简

知识的约简是在保持知识库中初等范畴的情况下，消除知识库中冗余的基本范畴，这一过程可以消去知识库中非必要的知识，仅仅保留真正有用的部分，即知识的“核”。

对于知识库可用知识表达系统形式化，知识库中任一等价关系在表中表示一个属性和用属性表示的关系的等价类。表中的列可以看作某些范畴的名称，而整个表包含了相应适应库中所有范畴的描述，能从表中数据导出的所有可能的规律，这就形成了一个决策表。通过这种表达，很容易用数据表的性质来表示知识库的基本性质，用符号代替语言定义，从而对知识的约简就变成对决策表的简化。

12.3.1　决策表的一致性

决策表中的对象 X 按条件属性与决策属性关系看作一条决策规则，可写成：

$$\wedge f_{C_i}(X) = f_D(X)$$

式中：C_i 表示多个条件属性；D 表示决定属性；f_{C_i} 表示对象 X 在 C_i 的取值；∧表示逻辑“与”。

对任一个对象，若条件属性有 $f_{C_i}(X_i) = f_{C_i}(X_j)$，则决策属性必须有 $f_D(X_i) = f_D(X_j)$，即一致性决策规则说明条件属性取值相同时，决策属性取值必须相同。

一致性决策规则也允许：若条件属性有 $f_{C_i}(X_i) \neq f_{C_i}(X_j)$，则决策属性可以是 $f_D(X_i) = f_D(X_j)$ 或 $f_D(X_i) \neq f_D(X_j)$。

在决策表中如果所有对象的决策规则都是一致的，则该信息表是一致的，否则信息表是不一致的。在进行属性约简时，每约掉一个属性时要检查属性表，若保持一致性，则可以删除，否则不可以删除。

12.3.2　属性约简

决策表中决策属性 D 依赖条件属性 C 的依赖度定义为

$$\gamma(C, D) = \frac{|\mathrm{POS}(C, D)|}{|\widetilde{X}|}$$

其中：｜POS（C,D）｜表示正域 POS（C,D）元素的个数，｜$\widetilde{X}$｜表示整个对象集合的个数。

γ（C,D）的性质如下：

① 若 $\gamma=1$，表示在已知条件 C 下，可以将 $\widetilde{X}$ 上全部个体分类到决策属性 D 的类别中去。

② 若 $\gamma=0$，即利用条件 C 不能分类到决策属性 D 的类别中去。

③ $0<\gamma<1$，即在已知条件 C 下，只能将 $\widetilde{X}$ 上那些属于正域的个体分类到决策属性 D 的类别中去。

设 C，$D\subset A$，C 为条件属性集，D 为决策属性集，$a\in C$，属性 a 关于 D 的重要度定义为

$$\mathrm{SGF}(a,C,D)=\gamma(C,D)-\gamma(C-\{a\},D)$$

式中：$\gamma(C-\{a\},D)$，表示在 C 中缺少属性 a 后，条件属性与决策属性的依赖程度；$\mathrm{SGF}(a,C,\ \ )$ 表示 C 中缺少属性 a 后，导致不能被正确分类的对象在系统中所占的比例。

$\mathrm{SGF}(a,C,\ \ )$ $\mathrm{SGF}(a,C,\ \ )$ 的性质：

① $\mathrm{SGF}(a,C,\ \ \in[0,1]$。

② $\mathrm{SGF}(a,C,\ \ )=0$，表示属性 a 关于 D 是可约简的。

③ $\mathrm{SGF}(a,C,\ \ )\neq0$，表示属性 a 关于 D 是不可约简的。

设 C、D 分别是信息系统 S 的条件属性和决策属性集，属性集 P（$P\subseteq C$）是 C 的一个最小属性集，当且仅当 γ（C,D）并且 $\forall P'\subset P,\gamma(P',D)\neq\gamma(P,D)$，说明若 P 是 C 的最小属性集，则 P 具有与 C 相同的区分决策类的能力。

12.3.3 分辨矩阵与分辨函数

决策表的分辨矩阵是一个对称的 n 阶方阵，其元素定义为

$$m_{ij}^{*}=\begin{cases}\{a\mid a\in C\text{且}f(x_i,a)\neq f(x_j,a)\} & (x_i,x_j)\notin IND(D)\\ \varnothing & (x_i,x_j)\in IND(D)\\ -1\quad f(x_i,a)=f(x_j,a)\} & \text{且}(x_i,x_j)\notin IND(D)\end{cases}$$

在构造决策表的分辨矩阵时要注意，只有在 x_i、x_j 不属于同一决策类的前提下，m_{ij}^{*} 是可以区分 x_i、x_j 的所有属性的集合；若 x_i、x_j 属于同一类决策类时，则分辨矩阵中元素 m_{ij}^{*} 为 $\varnothing$，而当所有属性值相同但决策类不同，即不符合一致性原则，元素值为 -1，表明数据有误或者提供的条件属性不足。

由于分辨矩阵是矩阵，在计算时只需写出分辨矩阵的下三角部分即可。

C 的 D 核是分辨矩阵所有单个元素 m_{ij}^{*} 的并，即

$$CORE_D(C)=\{a\in C\mid m_{ij}^{*}=\{a\}\quad 1\leqslant i,j\leqslant n\}$$

决策表的分辨函数定义为下式，即为元素的合取和析取。

$$\rho^{*}=\wedge\{\vee m_{ij}^{*}\}$$

12.4　模糊集与粗糙集

粗糙集与模糊集并非是对立的理论，两者既互相区别，又互相补充。从根本上讲，粗糙集体现了集合中对象间的不可区分性，即由于知识的粒度而导致的粗糙性；而模糊集则对集合中子类的边界的不清楚定义进行模型化，体现的是隶属边界的模糊性。它们处理的是两种不同的模糊和不确定性，如果将两者有机地结合在一起能更好地处理不完全的知识。

粗糙集与模糊集都是经典集合论的拓展，但它们之间有较多不同的地方。模糊集主要着眼于知识的模糊性，它是通过对象关于集合的隶属程度来近似描述模糊性，但其隶属函数一般是由专家给出，具有较强的主观性；而粗糙集则强调知识的粗糙性，它是通过一个集合上、下近似计算出来的而不是事先假定的，两者反映的知识粒度不同。从集合的对象间的关系来看，模糊集强调集合边界的状态，反映集合本身的含糊性，而粗糙集强调的是集合对象间的不可分辨性。从研究对象来看，模糊集研究的是属于同一类的不同对象对集合的隶属关系，重在隶属程度，因此模糊集是数据挖掘中常用的聚类方法之一；而粗糙集研究的是不同类中的对象组成的集合之间的关系，重在分类，分类的能力在于论域上的不可分辨关系提供的知识多少。

粗糙集理论的优势在于它不需要任何预务的额外的数据信息，则模糊集和概率统计等处理不确定的常用方法需要一些数据的附加信息或先验知识，如模糊隶属函数和概率分布等。但粗糙集也有其局限性，单纯地使用粗糙集理论不一定能完全有效地描述不精确或不确定的问题，因此在实际应用中，常将粗糙集理论与模糊集理论结合起来。这是因为这两者都是描述不精确事物的方法，只是侧重面不同。粗糙集主要用于处理区间值或一组值的情况，而模糊集主要用于将具有模糊意义的精确数据用模糊分段的方法详细描述的情况。

12.5　基于 MATLAB 的粗糙集处理方法

例 3.11　对于表 12.1 所示的决策表，求分辨矩阵和核。

表 12.1　决策表

U	*a*	*b*	*c*	*d*	*e*
X_1	1	0	2	1	1
X_2	1	0	2	0	1
X_3	1	2	0	0	2
X_4	1	2	2	1	0
X_5	2	1	0	0	2
X_6	2	1	1	0	2
X_7	2	1	2	1	1

解：

根据分辨矩阵的定义，可编程计算。分辨矩阵中单元素的元素项即为核。

```
>>x=[1 0 2 1 1;1 0 2 0 1;1 2 0 0 2;1 2 2 1 0;2 1 0 0 2;2 1 1 0 2; 2 1 2 1 1; 1 2 0 0 1];
>>y=core(x);y{2}='I2'    %即'b',程序中的属性用'Ii'表示
>> y{1}=[]    []           []           []           []         []          []   []
```

```
          []        []            []            []            []         []          []   []
   {1×3 cell}   {1×2 cell}   []            []            []         []          []   []
   {1×1 cell}   {1×2 cell}   {1×2 cell}    []            []         []          []   []
   {1×4 cell}   {1×3 cell}   []            {1×4 cell}    []         []          []   []
   {1×4 cell}   {1×3 cell}   []            {1×4 cell}    []         []          []   []
          []        []       {1×4 cell}    {1×2 cell}    {1×2 cell} {1×2 cell}  []   []
          []        []       [ -1]         {1x2 cell}    {1×2 cell} {1×3 cell}  []   []
```

例 3.12　对表 12.2 中的决策表进行约简。

表 12.2　决策表 *d*

U	*a*	*b*	*c*	
X_1	1	1	0	0
X_2	1	1	1	1
X_3	1	1	2	1
X_4	0	1	0	0
X_5	0	0	1	0
X_6	0	1	2	1
X_7	1	0	1	1
X_8	0	0	0	0

解：

对决策表的约简可以用两种方法：一是利用分辨矩阵，再利用逻辑运算就可以得到核及约简属性；二是根据属性的重要度确定可以约简的属性，即重要度为 0 的属性可以删除，但此时要检查一致性，即约简后的决策表不能存在相互矛盾的规则。

根据此原理，就可以编程对所给的决策表进行约简。

```
>> x=[1 1 0 0;1 1 1 1;1 1 2 1;0 1 0 0;0 0 1 0;0 1 2 1;1 0 1 1;0 0 0 0];
>> y=reduction_rough(x);
y{1}=redu: {'I2'}                            %可以约简的属性
 keep: {'I1'  'I3'}                          %决策有保留下的属性
dnum: 2                                      %可以约简属性的序号
```

由于在某些较为复杂决策表的约简中，可以有多种约简选择，所以规则的建立用另外的函数计算，其中建立后的规则，既可以进行简化，也可能维持不变。

```
>> y=rule_rough(x,2,'on');                   %on 表示对规则进行简化，off 表示不简化
>> y=rule: {[2×3 double]  [2×3 double]}      %最后形成的规则
    pro: {'I1'  'I3'  'd'}                   %最后形成决策表的表头
>> y.rule{1}= NaN  0  0                      %NaN 表示此属性在此规则中可以忽略
             0   1  0
>> y.rule{2}=1   1   1
```

```
        NaN  2  1
```

例 3.13　某机械常见故障有磨损、叶片断裂、动平衡破坏、同心度偏移、油膜失稳等。当发生这些故障时，会出现多种征兆，尤其以振动现象最为明显、普遍。通过研究该机械的故障振动表现为其旋转频率的倍频。因此，可以用该机械在这些频率成分上的振动能量作为特征信息来诊断识别各种故障。

通过分析测量得到表 3.21 所示的数据（已经离散化），其中属性分别用 x_1、x_2、x_3、x_4和 x_5表示，故障用 D 表示。试用粗糙集理论分析之。

表 12.3　某机械故障的决策表

样　本	x_1	x_2	x_3	x_4	x_5	D
1	3	1	3	1	2	1
2	1	3	2	1	3	2
3	3	1	2	3	1	3
4	2	3	2	1	3	2
5	1	3	1	1	3	2
6	3	1	3	2	2	1
7	3	1	3	2	2	1
8	1	1	3	2	1	1
9	3	1	2	3	3	3
10	2	1	3	3	1	3
11	1	2	2	3	2	3
12	1	3	1	1	3	2
13	2	1	3	2	3	1
14	1	1	3	2	2	
15	1	3	2	1	1	2
15	3	1	3	2	2	3

解：

```
>>load x;y=reduction_rough(×);
>> y{1}=redu: {3×2 cell}
        keep: {3×3 cell}
        dnum: [3×2 double]
>> y{1}.redu='I2'  'I3'
             'I2'  'I4'
             'I3'  'I4'
>> y{1}.keep='I1'  'I4'  'I5'
             'I1'  'I3'  'I5'
             'I1'  'I2'  'I5'
```

从结果中看出，本决策表的核是'I1'和'I5'，可以约简的属性为 x_2、x_3、x_4，但不能同时约简，否则有相矛盾的规则，即不符合一致性，因此，有 3 种约简方法，可以通过优化方法确定最优的约简方法。在此设定以下的约简，然后可以计算规则：

```
>> y=rule_rough(x,[2 3],'off');       %即删除 x2、x3 属性
>> y= rule: {[5×4 double]  [3×4 double]  [5×4 double]}
      pro: {'I1'  'I4'  'I5'  'd'}
```

根据以上的决策表作为训练集，利用人工神经网络等方法就可以判别不同情况下的该机械的故障种类。

例 3.14　某证券公司为了更好地提高对不同客户的服务质量，需要对客户分类。根据资金余额、总成交额、总成交量和交易频度 4 个指标，将客户确定为 VIP、IP 和 CP（由专家根据 4 个指标值的不同情况决定）。现根据相关数据得到表 12.4 的决策表。试求客户的分类方法。

表 12.4　决策表

样　本	x_1	x_2	x_3	x_4	D
1	2	1	3	2	2
2	2	3	1	2	3
3	3	2	3	3	1
4	2	1	2	2	2
5	3	2	2	3	2
6	3	2	1	2	3
7	2	2	3	2	1
8	2	3	3	2	1
9	3	2	1	3	2
10	3	2	3	2	2

解：

可以根据决策表求出每个指标的权重，然后根据每个客户具体这4个指标的数值，便可以求出客户的重要程度。

```
>> x=[2 1 3 2 2;2 3 1 2 3;3 2 3 3 1;2 1 2 2 2;3 2 2 3 2;3 2 1 2 3;2 2 3 2 1;2 3
     3 2 1; 3 2 1 3 2;3 2 3 2 2];
>>y=importance_rough(x,(1:4)');
```

从结果中，可看出各指标的重要性为0.2、0.3、0.700和0.40，相应的权重系数为0.1250、0.1875、0.4375和0.2500。由于第1、2个指标的权重基本相同，可以将其合并为一个，从而可得到决定客户重要性是的各指标比例为20%、70%和10%。

根据各指标的权重，便可在一定数据的基础上，对客户进行分类。

第 13 章

目标优化技术

13.1 目标优化概述

人们在科学实验、生产技术改进、工程设计、社会经济问题分析、管理决策等实际工作中，都倾向于采取某种措施，以便在有限的资源条件下或规定的约束条件下得到最满意的效果，这就引出了优化问题，即在满足一定的约束条件下，寻找一组参数值，以使系统的某些性能指标（目标函数）达到最大或最小。优化问题在工业、社会、经济、管理等各个领域都有广泛的应用，其重要性是不言而喻的。在数据挖掘的很多算法中，问题最终也常常归结为一个目标优化问题。

优化是指在合理的时间范围内为一个优化问题寻找可行解的过程，其中优化问题的可行解之间是可以进行量化比较的。寻优问题最优可行解过程的第一步是要对问题进行描述并在此基础上建立数学模型，即利用数学方程式和不等式来描述说明所求的优化问题，其中包括目标函数和约束条件，而识别目标、确定目标函数的数学表达形式尤为关键。优化问题根据目标函数、约束函数的性质以及优化变量的取值等可以分成多种类型，每一种类型的优化问题根据性质的不同都有其特定的求解方法。

不失一般性，优化问题可以描述为

$$\min \sigma = f(X)$$
$$\text{s.t.}\quad X \in S = \{X \mid g_i(X) \leqslant 0, i = 1, \cdots, m\}$$

式中：$\sigma = f(X)$ 为目标函数；$g_i(X)$为约束函数；S 为约束域；X 为 n 维优化变量。当 X 为连续变量时，最优化问题为函数优化问题；当 X 为离解变量时，最优化问题变为组合优化问题。

当 $f(X)$、$g_i(X)$为线性函数且 $X \geqslant 0$ 时，上述优化问题即为线性规划问题，其求解方法有成熟的单纯形法和卡马卡（Karmarkar）方法。

当 $f(X)$、$g_i(X)$中至少有一个函数为非线性函数时，上述问题即为非线性规划问题。非线性规划问题相当复杂，其求解方法多种多样，但到目前为止仍然没有一种有效的适合所有问题的方法。

当优化变量 X 仅取整数值时，上述问题即为整数规划问题，特别是当 X 仅能取 0 或 1 时，上述问题即为 0－1 规划问题。由于整数规划问题属于组合优化范畴，其计算量随变量维数的增长而呈指数增长，因此存在着维数灾难问题。

当 $g_i(X) \leqslant 0$（$i=1,2,\cdots,m$）所限制的约束空间为整个 n 维欧氏空间，即 $\boldsymbol{R}^n$ 时，上述优化问题就为无约束优化问题。

对于非线性规划问题，函数的非线性使得问题的求解变得十分困难，特别是当目标函数在约束域内存在多峰值时，常见的求解非线性问题的优化方法其求解结果与初值的选择关系很大。也即一般的约束或无约束非线性优化方法均是求问题函数在约束域内的近似极值点，而非真正的极值点。

优化问题的解包括全局最优解和局部最优解，有些优化问题，如 NP 问题（non-polynomial problem）只能取得局部最优解或次优解。

一般而言，优化问题都是一些难解问题，特别是随着非凸、非线性、高维、多变量、多模、多约束条件、多目标函数等复杂优化问题不断地被提出，优化问题也越来越复杂。在自然计算中，常用计算复杂性来描述问题的难易程度或算法的执行效率。算法的执行效率主要指算法执行时的时间消耗，包括运行时间开销和存储时间开销两个方面，前者称为算法的时间代价，后者称为算法的空间代价。

对于复杂性较高的优化问题，传统的算法求解往往不能进行有效的求解，或者求解的时间过长或求解的效果差而令人无法接受。对于这些问题，智能优化算法作为一种随机性优化算法，能够真正有效地解决以上问题，且具有一定的普适性。

13.2　极值问题

极值是在某个定义范围内函数的最大值或最小值。问题的对象可以有连续和离散两种情况，需要使用不同的方法对它们求解。

1. 连续情况

在数学上，极值的必要条件是函数 $f(X)$在 x_0 处的一阶导数等于 0，即

$$\left.\frac{df(x)}{dx}\right|_{x=x_0}=0$$

极大值和极小值的区分是根据函数在相应位置二阶导数取值的正负情况确定，即

极大值条件：$\left.\frac{d^2f(x)}{dx^2}\right|_{x=x_0}<0$

极小值条件：$\left.\frac{d^2f(x)}{dx^2}\right|_{x=x_0}>0$

根据定义，就可以求出连续函数的最优值。

2. 离散情况

对于离散情况的极值求解，与连续情况的类似，不同处为差分代替微分。

13.3　无约束非线性规划

对于一般的非线性函数 f，用解析法得到精确解比较困难，常用的方法是用搜索法求得近似最优解。无论搜索是在多维空间进行的，思路都是和一维空间搜索相同的。

类似于一维搜索，第 k+1 次迭代后 x 的位置为

$$x^{(k+1)}=x^{(k)}+\lambda^{(k)}d^{(k)} \qquad k=0,1,2,\cdots$$

其中：$x^{(k)}$为第 k 次迭代后 x 的位置；$\lambda^{(k)}$为第 k 次步长，$d^{(k)}$表示第 k 次搜索方向。

利用上式，逐步搜索，逐次逼近极小值。如果每一步有

$$f(x^{(k+1)})<f(x^{(k)})$$

则在一定次数的迭代后，会满足下面至少一个结果。第 1 种结果是，“搜索方向”的模量已经足够小

$$|\, d^{(k)} \,||<\varepsilon_1, 0<\varepsilon_1\,|$$

第 2 种结果是，每前进一步，目标函数几乎没有改进

$$|f^{(k+1)}-f(x^{(k)})\|<\varepsilon_2, 0<\varepsilon_2|$$

只要这两者中有一个已达到精度，就停止搜索，并将 $x^{(k+1)}$看作是近似极小点。

求解非线性多元函数极值的迭代法可以粗略地分为直接法和间接法。直接法只用到函数本身的信息，显见，由于用到的目标函数信息较少，收敛速度也慢一些。而间接法要用到函数的导数，由于用到的目标函数信息较多，收敛速度也较快。间接法也称为解析法，但只能用于目标函数有解析式、可求导数的场合。下面为常用的几种间接法。

13.3.1　梯度下降法

由于负梯度方向是函数值下降最快的方向，所以梯度下降法就是在迭代的每一步都沿着负梯度方向移动一段距离。

梯度下降法也称最速下降法，它对初始点的选取要求不严，迭代过程简单，便于使用。

设目标函数 $f(x)$具有一阶连续偏导数，且极值存在，其负梯度方向就是函数值 $f(x)$的最速下降方向。迭代公式为

$$x^{(k+1)}=x^{(k)}-\lambda^{(k)}\nabla f(x^{(k)}) \quad k=0,1,2,\cdots$$

其中$0\leqslant\lambda^{(k)}$称为步长或学习率，其计算公式为

$$\lambda^{(k)}=\frac{\nabla f(x^{(k)})^T\nabla f(x^{(k)})}{\nabla f(x^{(k)})^T H(x^{(k)})\nabla f(x^{(k)})}$$

其中 H 为 Hesse 阵

$$H(x^*)=\begin{bmatrix}\frac{\partial^2 f(x^*)}{\partial x_1\partial x_1} & \frac{\partial^2 f(x^*)}{\partial x_1\partial x_2} & \cdots & \frac{\partial^2 f(x^*)}{\partial x_1\partial x_N}\\ \frac{\partial^2 f(x^*)}{\partial x_2\partial x_1} & \frac{\partial^2 f(x^*)}{\partial x_2\partial x_2} & \cdots & \frac{\partial^2 f(x^*)}{\partial x_2\partial x_N}\\ \vdots & \vdots & \cdots & \vdots\\ \frac{\partial^2 f(x^*)}{\partial x_N\partial x_1} & \frac{\partial^2 f(x^*)}{\partial x_N\partial x_2} & \cdots & \frac{\partial^2 f(x^*)}{\partial x_N\partial x_N}\end{bmatrix}$$

梯度下降法的算法步骤如下。

（1）给定初始近似点 $x^{(0)}$及精度要求 $\varepsilon_1>0$，$\varepsilon_2>0$。如果$\|\nabla f(x^{(0)})\|^2\leqslant 0$，则停止，并令 $x^*=x^{(0)}$，得近似最小点 x^*；否则令 $k=0$，进行下一步。

（2）若$\|\nabla f(x^{(k)})\|\geqslant\varepsilon_2$，则可用一维搜索法、微分法计算最优搜索步长 $\lambda^{(k)}$，或者设定一个 $\lambda^{(k)}$试算，求

$$x^{(k+1)}=x^{(k)}-\lambda^{(k)}\nabla f(x^{(k)}) \quad k=0,1,2,\cdots$$

（3）如果$\|\nabla f(x^{(k)})\|\leqslant\varepsilon_1$或$\|f^{(k+1)}-f(x^{(k)})\|<\varepsilon_2$，即达到精度要求，则停止，并令 $x^*=x^{(k+1)}$，得近似最小点 x^*，否则返回第 1 步。

13.3.2　共轭梯度法

梯度下降法的最速下降特性只有在求梯度的点 $x^{(k)}$附近，所以，沿此方向前进的步长不应太大，否则达不到最速下降。为了改进这个缺点，可以采用共轭梯度法和牛顿法等，使之在远离极小点时，收敛较快，而当接近极值点时，也有较为满意的收敛速度。

设 x 和 y 为 $N \times N$ 的对称正定阵 A 共轭，即满足 $x^{T}Ay=0$。

在共轭梯度法中可以使用下面公式计算步长和迭代的每一个点：

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} d^{(k)} \\ \lambda^{(k)} = -\dfrac{(\nabla f(x^{(k)})^{T} d^{(k)}}{(d^{(k)})^{T} A d^{(k)}} \\ d^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta^{(k)} d^{(k)} \\ \beta^{(k)} = -\dfrac{(\nabla f(x^{(k+1)})^{T} \nabla f(x^{(k+1)})}{\nabla f(x^{(k)})^{T} \nabla f(x^{(k)})} \\ k = 0,1,2,\cdots \end{cases}$$

13.3.3　牛顿法

高维搜索的牛顿法的计算公式为

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} d^{(k)} \\ d^{(k)} = -(H(x^{(k)}))^{-1} \nabla f(x^{(k)}) \\ \min\limits_{\lambda^{(k)}} f(x^{(k)} + \lambda^{(k)} d^{(k)}) \\ k = 0,1,2,\cdots \end{cases}$$

具体步骤如下。

（1）设定初始点 $x^{(0)}$，及梯度允许误差 $\varepsilon > 0$。若 $\|\nabla f(x^{(0)})\|^{2} \leqslant \varepsilon$，则极小点 $x^{(0)}=x^{(0)}$，迭代停止。令 $\overline{H}^{(0)}=1$，求

$$d^{(0)} - \overline{H}^{(0)} \nabla f(x^{(0)})$$

沿 $d^{(0)}$ 方向进行一维搜索，求得最优步长 $\lambda^{(0)}$，得 $x^{(1)} = x^{(0)} + \lambda^{(0)} d^{(0)}$

（2）计算 $\overline{H}^{(k)}$。

（3）求：$x^{(k+1)} = x^{(k)} + \lambda^{(k)} d^{(k)}$

其中：$d^{(k)} = -\overline{H}^{(k)} \nabla f(x^{(k)})$。

13.4　有约束非线性规划

典型的有约束非线性规划问题如下。

$$\min f(x)$$
$$\text{s.t.}\begin{cases} g_j(X) \geqslant 0 & j=1,2,\cdots,q \\ h_p(X)=0 & p=1,2,\cdots,m \\ x_i^l \leqslant x_i \leqslant x_i^u & i=1,2,\cdots,n \end{cases}$$

若上述目标或约束条件中，有一个或多个函数是非线性的，则此问题就称为非线性规划。

对于约束优化问题，一类重要的求解方法就是通过解一系列无约束优化问题以获取原非线性约束问题解的惩罚函数方法，其基本思想是：根据约束的特点构造某种惩罚函数，并把惩罚函数加到目标函数上，从而得到一个增广目标函数，使约束优化问题的求解转化为一系列优化问题的求解。故称此类算法为序列无约束极小化方法（sequential unconstrained minimization technique，SUMT）。常用的 SUMT 方法有两种，即外点法和内点法。

外点法的惩罚策略是：对违反约束条件的点在目标函数中加入相应的惩罚，而对可行点不予惩罚，其迭代点一般在可行域外部移动。随着迭代的进行，惩罚也逐次加大，以迫使迭代点不断逼近并最终成为可行点，以便找到原约束优化问题的最优解。

内点法的惩罚策略是：从一切可行点开始迭代，设法使迭代过程始终保持在可行域内部进行。为此，在可行域的边界设置一道“墙”。对企图穿越这道“墙”的点，在目标函数中加入相应的障碍，越接近边界，障碍就越大，从而就保证迭代点始终在可行域内部进行迭代。

采用惩罚函数外点法来优化约束问题时，增广目标函数可以表示为

$$\min F(x,\sigma)=f(x)+\sigma p(x)$$
$$p(x)=\sum_{i=1}^{m}[\max\{0,-g_i(x)\}]^{\alpha}+\sum_{j=1}^{l}|h_j(x)|^{\beta}$$

式中：$f(x)$ 为原函数；σ 为惩罚因子，是一个很大的正数；$p(x)$为惩罚函数。一般地，$\alpha=\beta=2$。

13.5 大规模优化问题的分解算法

在实际应用中往往是变量数和约束数都相当大。从理论上讲，只要线性规划和非线性规划是有限维的，使用一定的方法总是可以解的。但是这样的代价是出现所谓的“维数灾难”的问题。如果能将原问题分解成若干个“子问题”，先分别计算这些变量少、约束少的子问题，然后再综合考虑它们之间的关联，就可以从总体上大大减少机时。

13.5.1 问题的描述

设系统被划分成 C 个子系统。对第 i 个子系统（i=1,⋯,C），u_i 为对总系统也是对第 i 个子系统的输入，x_i 为由其他子系统提供的中间输入，v_i 为对第 i 个子系统的控制变量，y_i 为子系统的输出，z_i 是子系统 i 的输出，子系统 $j\neq i$ 的输入。以上各个向量，分别具有维数 $m_{u_i}, m_{x_i}, m_{v_i}, m_{y_i}, m_{z_i}$。

对于一个给定的总系统输入向量 u，子系统可用下述向量方程描述：

$$\begin{cases} z_i = g_i(v_i, x_i) \\ y_i = h_i(v_i, x_i) \end{cases}$$

子系统之间的联系为

$$x_i=\sum_{j=1}^{C}a_{ij}z_j, i=1,2,\cdots,C$$

其中：a_{ij} 为 $m_{x_i}\times m_{z_i}$ 矩阵，表达了子系统之间的耦合。

根据拉格朗日法原理，可得出最优解应满足的必要条件。

$$\begin{cases}\dfrac{\partial L}{\partial x_i}=\dfrac{\partial f_i(v_i,x_i)}{\partial x_i}+\dfrac{\partial g_i}{\partial x_i}\mu_i+\rho_i=0\\ \dfrac{\partial L}{\partial v_i}=\dfrac{\partial f_i(v_i,x_i)}{\partial v_i}+(\dfrac{\partial g_i}{\partial v_i})^T\mu_i=0\\ \dfrac{\partial L}{\partial z_i}=-\mu_i-\sum_{j=1}^{C}a_{ij}\rho_j=0\\ \dfrac{\partial L}{\partial \mu_i}=g_i-z_i=0\\ \dfrac{\partial L}{\partial \rho_i}=x_i-\sum_{j=1}^{C}a_{ij}z_j=0\end{cases}$$

上述方程组形成了两组递阶结构的分解协调算法。在该算法中，上级和下级之间不断交换信息，下级子系统向上级送出反馈变量，上级协调器根据各子系统来的反馈变量，从全局优化角度出发向下级给出协调变量，进行优化迭代，最后到总系统的最优点，如图 13.1 所示。

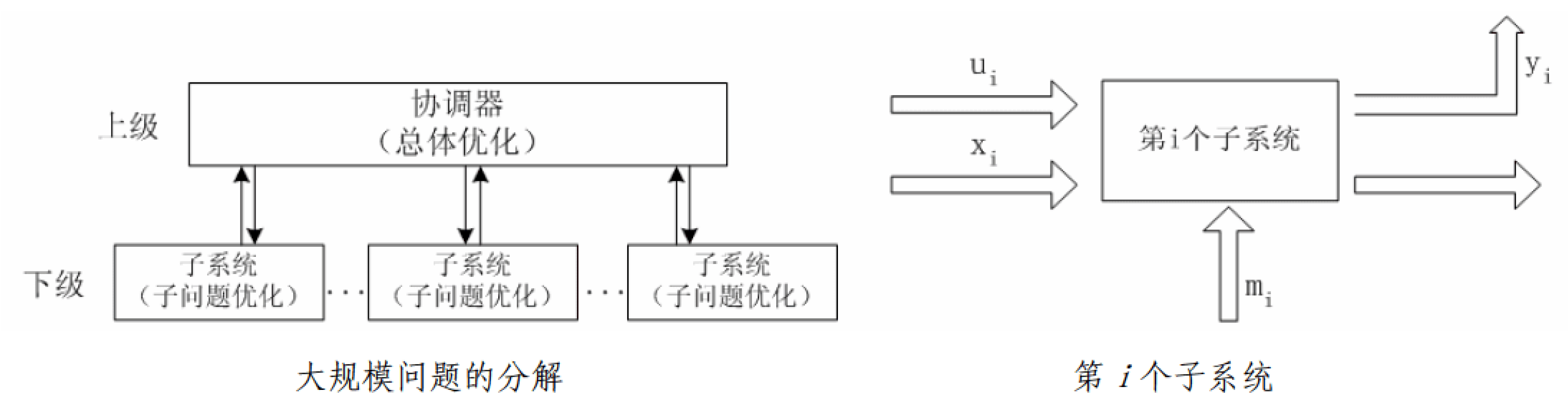


图 13.1　大规模优化问题

上述变量中，可以采用不同的变量做协调作用。相应地，形成了不同的分解协调算法，下面即为几种算法。

13.5.2　目标协调法

以拉格朗日乘子 ρ_i（i=1,2,⋯,N）为协调变量，x_i 和 z_i 为反馈变量，算法结构见图 13.2 所示。

设原问题为极大化问题，则子系统的子问题可写成

$$\begin{cases}\max[f_i(v_i,x_i)+\rho_i^T x_i-\sum_{j=1}^{C}\rho_j a_{ji}z_i]\\ z_i=g_i(v_i,x_i)\end{cases}$$

子问题拉格朗日函数为

$$L_i=f_i(v_i,x_i)+\mu_i^T(g_i-z_i)+\rho_i^T x_i-\sum_{j=1}^{C}\rho_j^T a_{ji}z_i$$

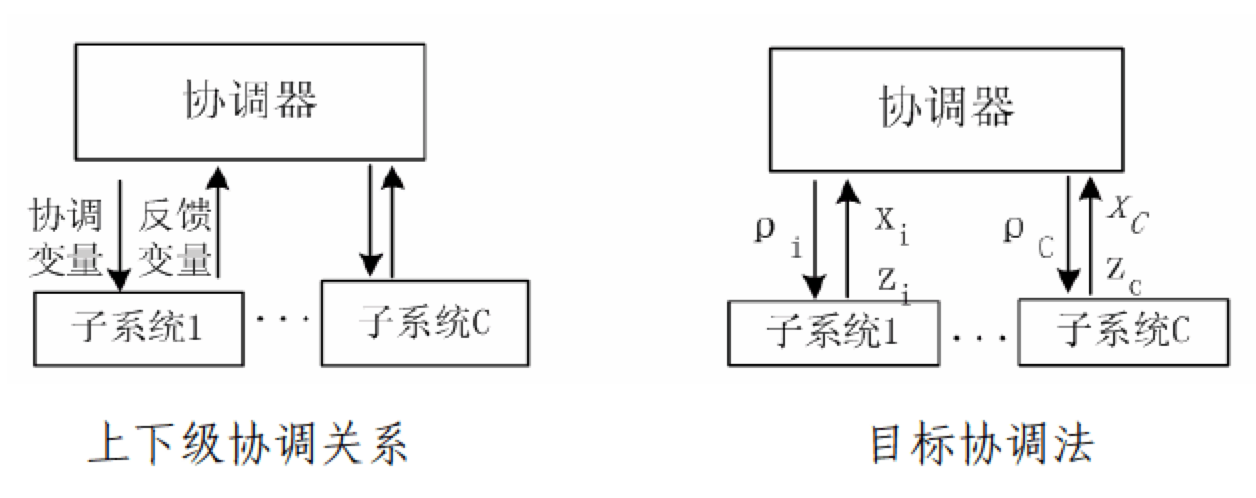


图 13.2　目标协调法

令 $x_i^*(\rho)$, $z_i^*(\rho)$, $v_i^*(\rho)$ 为子问题的解。求解第一级子问题的步骤如下：

（1）由上级给定协调变量值 ρ_i（i=1,2,⋯,N），求 μ_i；

（2）由 ρ_i、μ_i，求 x_i、v_i；

（3）由 x_i、v_i，求 z_i；

（4）由 $x_i^*(\rho)$, $z_i^*(\rho)$ 反馈到上级。

13.5.3　模型协调法

取关联变量 z_i 作为协调变量，使用拉格朗日乘子 ρ_i 和 μ_i 为反馈变量。

下级子系统的子问题可写成

$$\begin{cases} \max f_i(v_i,x_i) \\ z_i = g_i(v_i,x_i) \\ x_i = \sum_{j=1}^{c} a_{ij} z_j \end{cases}$$

拉格朗日函数可写作

$$L_i = f_i(v_i,x_i) + \mu_i^T[g_i(v_i,x_i) - z_i) + \rho_i^T(x_i - \sum_{j=1}^{C} a_{ji} z_i)$$

求解问题的步骤如下。

（1）由上级给定协调变量，即关联量的预估值 z_i（i=1,2,⋯,C），求 x_i；

（2）由 z_i、x_i，求 v_i、μ_i；

（3）由 z_i、x_i 和 v_i、μ_i，求 ρ_i，然后将 μ_i 和 ρ_i 反馈给上级。

13.5.4　混合协调法

将模型协调法和目标协调法相结合，在下级取 ρ_i 和 z_i 协调变量，给定后送给上级；上级以 x_i 和 μ_i 以反馈变量。

在给定 z 和 ρ 后，下级子问题可写作

$$\begin{cases} \max[f_i(v_i,x_i) + \rho_i^T(x_i - \sum_{j=1}^{C} a_{ij} z_j)] \\ z_i = g_i(v_i,x_i) \end{cases}$$

相应地，子问题拉格朗日函数为

$$L_i = f_i(v_i, x_i) + \mu_i^T [g_i(v_i, x_i) - z_i) + \rho_i^T (x_i - \sum_{j=1}^{C} a_{ji} z_i)$$

子问题的解法步骤如下：

（1）在上级给定协调变量 ρ_i 和 z_i（i=1,2,⋯,C），求 x_i 和 v_i；

（2）由 ρ_i 和 z_i，求 x_i、v_i、μ_i，并将 x_i 和 μ_i 反馈给上级。

13.6　其他优化方法

随着应用和需求的不断扩展，优化算法理论的研究也得到了长足的发展。就优化机制与行为来讲，目前工程中常用的优化算法除了经典算法外，出现了许多其他优化算法如构造型算法、改进型算法、基于系统动态演化的算法、混合型算法和群体智能算法等。

（1）构造型算法。用构造的方法快速建立问题的解，这种算法的优化质量通常较差，难以满足工程需要。例如调度问题中的典型构造型方法有 Johnson 法、Palmer 法、Gupta 法等。

（2）改进型算法，或称邻域搜索算法。从任一解出发，通过对其邻域的不断搜索和对当前解的替换来实现优化。根据搜索行为，其又可分为局部搜索法和指导性搜索法。

① 局部搜索法。利用局部优化策略在当前解的领域中贪婪搜索，如只接受优于当前解的状态作为下一个当前解的爬山法；接受当前解领域中的最好解作为一个当前解的最陡下降法。

② 指导性搜索法。利用一些指导规则来指导整个解空间中优良解的探索，如模拟退火算法、文化算法、差分进行算法、遗传算法、蚁群算法等各种群体智能算法。

（3）基于系统动态演化的方法。将优化过程转化为系统动态的演化过程，基于系统动态演化来实现优化，如神经网络和混沌搜索等。

（4）混合型算法。将上述各算法从结构或操作上进行混合而产生的各类算法，如文化基因算法等。

鉴于实际工程问题的大规模、强约束、非线性、从极值、多目标、建模困难等特点，寻找一种适合于大规模问题的具有智能特征的并行算法已成为相关学科的主要研究目标和引人注目的研究方向。

近 20 年来，一些新颖的优化算法，如人工神经网络、混沌、遗传算法、进化规划、模拟退火、禁忌搜索及其混合优化策略等，通过模拟或揭示某些自然现象或过程而得到发展，其思想和内容涉及数学、物理、生物进化、人工智能、神经科学和统计力学等方面，为解决复杂问题提供了新的思想和手段。这些算法的独特优点和机制，引起了国内外学者的广泛重视，并掀起了该领域的研究热潮，且在诸多领域得到了成功应用。近些年来，随着人工智能和人工生命的兴起，出现了一些新型的仿生算法，其中较具代表性的有蚁群算法、粒子群算法和人工鱼群算法。这些算法的产生为优化问题的解决提供了新的思路，更加推动了群体智能优化研究的发展。

值得指出的是，对于所有函数集合，并不存在万能的最佳优化算法，所有算法在整个函数类上的平均表现度量是相同的，关于优化算法的研究应从寻找所有可能函数类上的通用优化算法转变，包括以下两个方面：

（1）以算法为导向，确定其适用的问题类。对于每一个算法，都有其适用的和不适用的问题：对于给定的算法，要尽可能通过理论分析和实际应用，找出其适用的范围，归纳特定的问题

类，使其成为一个指示性算法。

（2）以问题为导向，确定其适用的算法。对于较小的特定问题类或特定的实际应用问题，设计出具有针对性和适用的算法。实际上，大多数在优化算法方面的研究都属于这一范畴。

13.7 基于 MATLAB 的目标优化方法

例 3.15 求函数 $f(x)=\mathrm{e}^{-x}\sin x^2$ 在[0,5]区间上的极大值和极小值。

解：

此函数的图像及其导数的图像如图 13.3 所示，其中左边图像是函数原像，右边是一阶、二阶导数的图像。从图中可看出，此函数在指定的区间中有多个极大值和极小值。

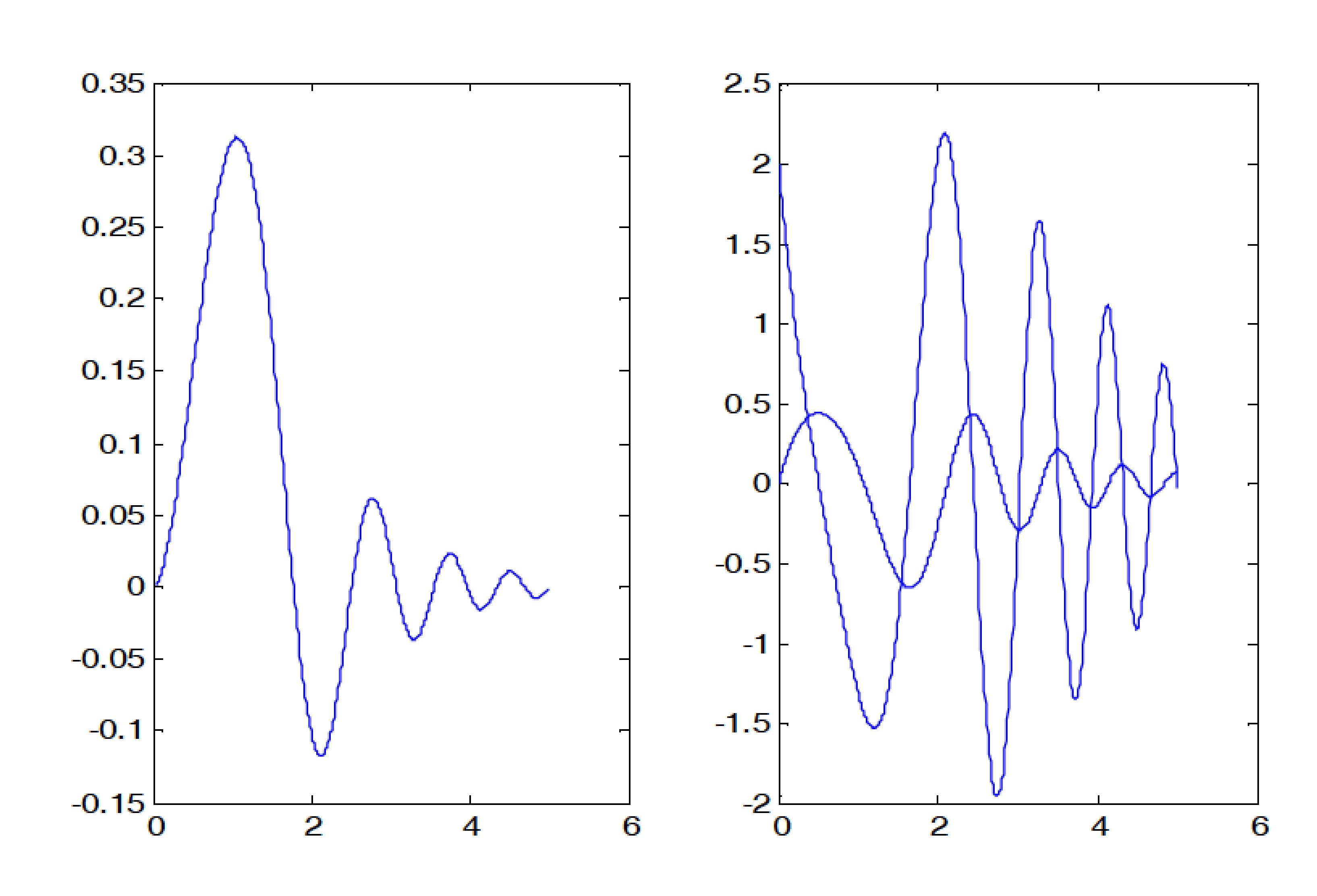


图 13.3 函数图像及其导数的图像

在 MATLAB 中，可以用 fzero 和 fsolve 求出方程的根，但是这两个函数都是计算给定初值附近的根，无法直接计算整个区间内所有的根。

解决方法是在整个区间上等间距地取多个取样点，即 $x_1,x_2,\cdots,x_n$，然后计算出相应的函数值 $f(x_1),f(x_2),\cdots,f(x_n)$。对上述各点逐一判断，计算函数 $F(x)=f(x_{k-1})f(x_k)$，$2\leqslant k\leqslant n$ 是否非正。如果 $F(x)\leqslant 0$，那么就用函数 fzero 计算 x_k附近的根，这样通过多次调用函数 fzero 就可以得到整个区间上所有的根。

```
>> syms x
>>fx=exp(-x)*sin(x^2);ds=diff(fx);d2s=diff(fx,2);
>> f1=@(x)exp(-x).*sin(x.^2);
>> f2=@(x)-exp(-x).*sin(x.^2)+2.*exp(-x).*cos(x.^2).*x;
>>f3=@(x)exp(-x).*sin(x.^2)-4*exp(-x).*cos(x.^2).*x-4*exp(-x).*sin(x.^2).*x.^2...
     …+2*exp(-x).*cos(x.^2);
```

```
>> x=0:0.010:5;y1=f1(x);subplot(121); plot(x,y1);
>> subplot(122);y2=f2(x);plot(x,y2);hold on;y3=f3(x);plot(x,y3);
>>ds=char(ds);                  %把 ds 的类型从符号型变为字符串型
>> ds=strrep(ds,'*','.*');      %用点乘代替乘号
>> ds=strrep(ds,'^','.^');      %用运算".^"代替乘号"^"
>> x0=fzeros(inline(ds),[0 5]);
>> d2f=subs(d2s,x0);
>> xM=x0(d2f<0)
 xM=1.0637  2.7705   3.7422  4.5066
>> xm=x0(d2f>0)
 xm=0   2.1167  3.2932  4.1423  4.8435
```

例 3.16　计算下列函数在 $x,y\in[-3\ 3]$的极值：

$$z = f(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10(\frac{x}{5} - x^3 - y^5)e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2}$$

解：

与一级导数的求解类似，计算二阶导数需要同时考虑两个自变量方向的一阶导数，当它们同时为零时，也就对应着极值点。可以通过极大值还是极小值判断零点与周围邻近点的大小关系。

在计算本题时，先用 jacobian 函数计算 $f(x,y)$的 5 个偏导数，然后用函数 fsolve 解出方程组的全部根（此时应将导数函数式转化为字符串型变量），继而得到相应驻点坐标。下面给出其中的部分代码：

```
>>syms xy;
>>z=3*(1-x).^2.*exp(-(x.^2)-(y+1).^2)-10*(x/5 - x.^3 -
y.^5).*exp(-x.^2-y.^2) ...
  - 1/3*exp(-(x+1).^2 - y.^2);
>>dF=jacobian(z,[x,y]);S1=char(dF(1));S2=char(dF(2));S1=strrep(S1,'*','.*');
>>S1=strrep(S1,'^','.^');S2=strrep(S2,'*','.*');S2=strrep(S2,'^','.^');
>>fun=['[',S1,',',S2,']'];fun=strrep(fun,'exp','q');fun=strrep(fun,'x','x(1)'
);fun=strrep(fun,'y','x(2)');
fun=strrep(fun,'q','exp');options=optimset('fsolve');options.TolFun=1e-8;
```

例 3.17　求下列向量型离散数据的极值：X=[8 2 10 7 4 3 6 9 5 1]。

解：

对于离散情况，可以利用差分代替微分来求解极值。MATLAB 中的 diff 函数既可以计算函数式的微分式，也可以用来计算差分。在计算差分后，如果相邻 3 个点的中间一点同时大于（或者小于）两侧的点，那么这一点就是极大值（或极小值）。

```
    >> x=[8 2 10 7 4  3 6 9 5 1];
    >> y=mymaxmin(x);        %差分法求极值函数
```

```
>> y=max: [10 9]  min: [2 3]
```

例 3.18　用最速下降法求解下列无约束非线性规划问题：

$$\min f = x_1^2 + 25x_2^2 \qquad x^0 = (2,2)^T$$

解：

根据最速下降法的原理，可编程计算：

```
>> [f0,x]=zuisuprog([2;2])
f0=1.0389e-013        x=1.0e-006 *[-0.3223,-0.0000]
```

例 3.19　用牛顿法求解下列非线性规划问题：

$$\min f = x_1^4 + 25x_2^4 + x_1^2 x_2^2 \qquad x^0 = (2,2)^T$$

解：

根据牛顿法的原理，可编程计算，其中为了提高计算精度，在迭代时采用变步长方法。

```
>> [x,f0]=newtonprog([2;2])                    %最速下降法函数
x= -0.0000  -0.0016      f0=1.5482e-010      %计算结果
```

例 3.20　在无约束非线性规划方法中，遇到问题的目标函数不可导或难以表达导函数的解析式时，人们一般需要使用直接搜索方法。同时，由于这些方法一般都比较直观和易于理解，因而在实际应用中常为人们所采用。这些方法中较为典型的便是 Powell 方法。请用此法，求下列函数的最小值：

$$\min f = 10(x_1 + x_2 - 5)^4 + (x_1 - x_2 + x_3)^2 + (x_2 + x_3)^6 \qquad x^0 = (0,0,0)^T$$

解：

根据powell算法原理，可编程计算。

先编写以下函数以求$f(x_1,x_2,x_3)$的函数值，然后利用powell函数求极值。由于在计算过程中，涉及符号计算，所以计算速度较慢，需迭代31次，才能得到结果。

```
function y=my_fun1(a)
syms x1 x2 x3
f=10*(x1+x2-5)^4+(x1-x2+x3)^2+(x2+x3)^6;
y=subs(f,{x1,x2,x3},a);
```

然后在 MATLAB 的工作窗口，输入下列命令：

```
>> x0=[0 0 0];y=powell(x0);
>> y=x: [3.3333 1.6667 -1.6667]  val: 1.5827e-031
```

例 3.21　在 MATLAB 工具箱中，用于求解无约束极值问题的函数有 fminunc 和 fminsearch。一般来说，当所选函数高度不连续或者变化剧烈时，使用 fminsearch 较好，而当函数中的变量幂次大于时，使用 fminunc 要比 fminsearch 有效。fminunc 的基本命令是：

[X,Fval]=fminunc(Fun,X0,Options,P1,P2,⋯)

其中的返回值 x 是所求得的极小点，fval 是函数的极小值。fun 是一个 m 文件，当 fun 只有一个返回值时，它的返回值是函数 $f(x)$；当 fun 有两个返回值时，它的第二个返回值是 $f(x)$的梯度向量；当 fun 有三个返回值时，它的第三个返回值是 $f(x)$的二阶导数阵（Hessian 阵）。x0 是向量 x 的初始值，options 是优化参数，可以使用默认参数。p1，p2 是可以传递给 fun 的一些参数。

求下列函数的最小值：$\min f = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

解：

编写求函数值及导数值的函数 fun1：

```
function [f,g]=fun1(x);
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
g=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
```

然后，在工作窗口输入命令，即可求得函数的极小值。

```
>> options = optimset('GradObj','on');[x,y]=fminunc('fun1',rand(1,2),options);
>> x=1.0000  1.0000
>> y=1.2424e-018
```

在求极值时，也可以利用二阶导数，此时需编写求一级及二级导数值的函数，然后利用 fminunc 就可求得最值。

```
function [f,df,d2f]=fun3(x);
 f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
 df=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
 d2f=[-400*x(2)+1200*x(1)^2+2,-400*x(1)-400*x(1),200];
>> options = optimset('GradObj','on','Hessian','on');
>> [x,y]=fminunc('fun3',rand(1,2),options);
```

例 3.22　带有约束条件的极值问题称为约束极值问题，也叫规划问题。求解约束极值问题要比求解无约束极值问题困难得多。为了简化其优化工作，可采用以下方法：将约束问题化为无约束问题；将非线性规划问题化为线性规划问题，以及能将复杂问题变换为较简单问题的其他方法。

求解下列二次规划问题。

$$\min f(x) = 2x_1^2 - 4x_1x_2 + 4x_2^2 - 6x_1 - 3x_2$$

$$\text{s.t.}\begin{cases} x_1 + x_3 \leqslant 3 \\ 4x_1 + x_2 \leqslant 9 \\ x_1, x_2 \geqslant 0 \end{cases}$$

解：

在 MATLAB 中，求解二次规划的是 Quadprog 函数，其基本用法如下：

```
[X,Fval]= Quadprog(H,F,A,B,Aeq,BEQ,Lb,Ub,X0,Options)
```

对于本例有：

```
>>h=[4,-4;-4,8];f=[-6;-3];a=[1,1;4,1];b=[3;9];[x,value]=quadprog(h,f,a,b,[],[],zeros(2,1));
>> x=1.9500  1.0500    value=-11.0250
```

例 3.23　求下列非线性规划：

$$\min f(x) = x_1^2 + x_2^2 + 8$$
$$\text{s.t.}\begin{cases} x_1^2 - x_2 \geqslant 0 \\ -x_1 - x_2^2 + 2 = 0 \\ x_1, x_2 \geqslant 0 \end{cases}$$

解：

利用罚函数法，可将非线性规划问题的求解转化为求解一系列无约束极值的问题。其基本思想是，利用问题中的约束函数作出适当的罚函数，由此构造出带参数的增广目标函数，把问题转化为无约束非线性规划问题。它主要有两种形式：一种叫外罚函数法；另一种叫内罚函数法。常见的是外罚函数法，它是将下列优化问题

$$\min f(x)$$
$$\text{s.t.}\begin{cases} g_i(X) \leqslant 0 & i = 1,2,\cdots,r \\ h_j(X) \geqslant 0 & j = 1,2,\cdots,s \\ K_m(x) = 0 & m = 1,2,\cdots,t \end{cases}$$

取一个充分大的数$M>0$，构造函数：

$$P(x,M) = f(x) + M\sum_{i=1}^{r}\max(g_i(x),0) - M\sum_{i=1}^{s}\min(h_i(x),0) + M\sum_{i=1}^{t}|K_i(x)|$$

对于本题，先编写如下test函数：

```
 function g=test(x)
M=50000;f=x(1)^2+x(2)^2+8;
g=f-M*min(x(1),0)-M*min(x(2),0)-M*min(x(1)^2-x(2),0)+...
  M*abs(-x(1)-x(2)^2+2);
%g=f-M*sum(min([x';zeros(1,2)]))-M*min(x(1)^2-x(2),0)+...    %另一种表示形式
%   M*abs(-x(1)-x(2)^2+2);
%g=f-M*min(min(x),0)-M*min(x(1)^2-x(2),0)+...                %另一种表示形式
%   M*(-x(1)-x(2)^2+2)^2;
```

在MATLAB工作窗口输入命令：

```
>> [x,y]=fminunc('test',rand(2,1))
x=1.02860.9856  y=10.0297
```

例 3.24　求函数族$\{f_1(x), f_2(x), f_3(x), f_4(x), f_5(x)\}$取极大极小值时的$x$值，其中：

$$\begin{cases} f_1(x) = 2x_1^2 + x_2^2 - 48x_1 - 40x_2 + 303 \\ f_2(x) = -x_1^2 - 3x_2^2 \\ f_3(x) = x_1 + 3x_2 - 18 \\ f_4(x) = -x_1 - x_2 \\ f_5(x) = x_1 + x_2 - 8 \end{cases}$$

解：

在MATLAB中，求极大极小值的函数是fminimax，它可以解决如下形式的规划问题：

$$\min_{x}\{\max_{F_i} F(x)\}$$

$$\text{s.t.}\begin{cases} A*x \leqslant b \\ Aeq*x = Beq \\ C(x) \leqslant 0 \\ Ceq(x) = 0 \\ LB \leqslant x \leqslant UB \end{cases}$$

其基本用法如下：X=fminimax(Fun,X0,A,B,Aeq,Beq,Lb,Ub,Nonlcon)

先编写以下函数：

```
function  f=fun8(x)
f=[2*x(1)^2+x(2)^2-48*x(1)-40*x(2)+304
  -x(1)^2-3*x(2)^2
   x(1)+3*x(2)-18
   -x(1)-x(2)
   x(1)+x(2)-8];
```

在MATLAB工作窗口中，输入命令便可以做到：

```
>> [x,y]=fminimax(@fun8,rand(2,1))
x=4.0000    4.0000
y=     0
 -64.0000
  -2.0000
  -8.0000
       0
```

例 3.25　在求解最优问题时，除用命令行方法外，还可以利用 MATLAB 优化工具箱中的 optimtool 的 GUI 方法。optimtool 可应用到所有优化问题的求解，计算结果可以输出到 MATLAB 工作空间中。在 MATLAB 的较高版本中，此 GUI 中的优化方法（Solve）不仅包含了统计工具箱中的各种优化函数，而且还提供遗传算法、模拟退火等优化方法。

请利用优化问题的 GUI 方法求解下列非线性规划问题：

$$\min f(x)=x_1^2+x_2^2+x_3^2+8$$

$$\text{s.t.}\begin{cases} x_1^2-x_2+x_3^2\geqslant 0 \\ x_1+x_2^2+x_3^3\leqslant 20 \\ -x_1-x_2^2+2=0 \\ x_2+2x_3^2=3 \\ x_1,x_2,x_3\geqslant 0 \end{cases}$$

解：

首先编写目标函数文件和约束条件函数文件：

```
function f=fun4(x)                              %目标函数
 f=sum(x.^2)+8;
function [g,h]=fun5(x)
g=[-x(1)^2+x(2)-x(3)^2;x(1)+x(2)^2+x(3)^3-20];  %非线性不等性约束
   h=[-x(1)-x(2)^2+2;x(2)+2*x(3)^2-3];          %非线性等式约束
```

然后在MATLAB命令窗口运行optimtool，就打开图形界面，如图13.4所示，填入有关的参数，未填入的参数取值为空或者为默认值，然后用单击start按钮，就得到求解结果，再使用file菜单下的Export toWorkspace…命令，把计算结果输出到MATLAB工作空间中去。此例中以a（它是一个结构体）代表求解结果。

```
>> a.x=0.55221.20330.9478          %函数最小值时的各变量值
>> a.fval=10.6511                  %函数最小值
```

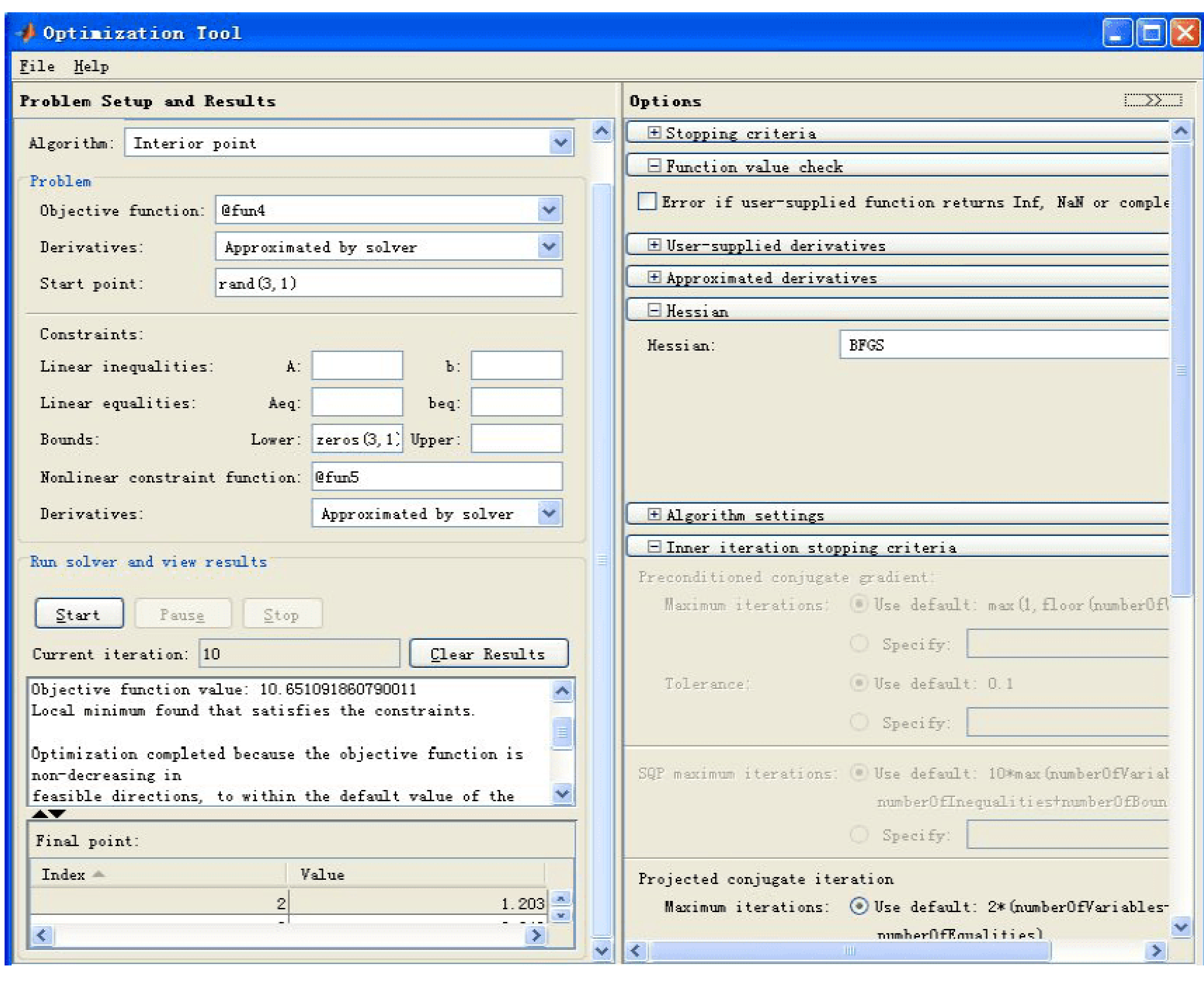


图 13.4　求解优化问题的 GUI

第 14 章

可视化技术

14.1 可视化技术概述

计算机科学和技术的进展在科学工程和商业领域导致了许多不可预测的可能性。与此同时，测量的自动化、网络传感、过程的数字化和大量的计算机仿真产生了海量的数据，数据的增长量超过人分析理解的能力。可视化提供了一种解决这类问题的新工具。

一般意义下的可视化定义为：可视化是一种使复杂信息能够容易和快速被人理解的手段，是一种聚集在信息重要特征的信息压缩语言，是可以放大人类感知的图形化表示方法。

可视化技术也称数据可视化，它旨在凭借计算机的强大信息处理能力以及计算机图形学基本算法及可视化算法将计算机进行的大规模科学（工程）计算结果及其产生的数字数据、信息和知识转换成静态或动态图像的过程，并允许人们通过交互手段控制数据的抽取和画面显示，并且获得对数据更深层次的认识。它具有以下的特点：

- 交互性。用户可以方便地以交互的方式管理和开发数据；
- 多维性。可以表示对象或事件的数据的多个属性或变量；
- 可视性。数据可以用图像、曲线、二维图形、三维图形和动画显示，并可对其模式和相关关系进行可视化分析。

数据可视化技术对于大型数据集的分析及浏览有着非常重要的作用，它可以大大加快数据处理速度，特别是在数据挖掘运行过程中，可视化技术可以给用户提供交互操作，并可以从中为用户反馈重要信息。尤其在用户对数据描述知之甚少、对挖掘目的不明确的情况下更为有效。例如利用可视化技术对环境污染的传播、全球臭氧分布、建筑物与周围气流、大面积水域污染等问题进行模拟、试验，分析产生的结果，可为人类在环境生态学方面提供切实可行预报措施；利用可视化技术在地质勘探中，利用自然地震波或人工爆破产生的声波在不同地质构造层中的传播速度和衰减程度的不同特点，利用反演变换重构表示地质结构的体数据，以帮助寻找新的矿产，并确保以发现矿产的最佳状态，取得良好的经济效益。

可视化数据挖掘不仅局限于用图形图像表现数据，还要能够发现其中隐含的信息和知识。运用数据可视化技术不仅能够展现数据挖掘过程得到的数据，还能补充数据挖掘过程，增加对数据挖掘算法的理解。通过在数据挖掘过程中使用可视化技术。

- 能够在挖掘过程中随时剔除异类和噪声数据，提高挖掘质量；
- 能够利用人类的模式识别能力评估和提高挖掘出的结果模式的有效性；
- 建立用户与数据挖掘系统交互的良好沟通通道，能够使用户利用专业背景来约束挖掘，不需要具备复杂的数学和统计学知识，改善挖掘结果；
- 通过对数据挖掘结果的可视化，使用户获得结果模式的直观理解，打破传统挖掘算法的黑盒模式，使用户对挖掘系统的依赖程度大大提高。

可视化技术与数据挖掘主要从以下几个方面相结合，形成可视化数据挖掘。

（1）数据可视化：在进行数据挖掘算法之前对数据进行可视化研究，将数据库或者数据仓库中的数据，从不同粒度和不同的抽象层次或将属性、维度进行联合之后，把数据表转换为可视化结构，并以各种不同类型的形式展现在用户的面前。

（2）数据挖掘过程可视化：这种方法将可视化技术融入到数据挖掘过程中，在交互式的可视化数据挖掘中使用可视化工具，用户可以通过设置参数来控制整个挖掘过程的进度和质量，并且依

靠感觉、具备的领域知识以及挖掘算法的结果共同做出决策，是最理想的可视化数据挖掘过程。

（3）数据挖掘结果可视化：将挖掘后得到的知识或者结果用可视化的形式表示出来，使原本抽象的挖掘结果信息简明化，加速人们对结果信息的特征、关系、模式和趋势等的理解，从而对挖掘结果的正确性作出判断，得出科学的挖掘结果。用户可以根据结果信息，迭代的调整算法的参数，使得挖掘结果更符合人们的需求。

14.2　可视化技术分类

根据可视化对象的不同，可视化技术可分成以下四类。

14.2.1　数据可视化

数据可视化是运用计算机图形学和图像处理技术，将数据转换为图形或图像在屏幕上显示出来，并进行交互处理的理论、方法和技术。

数据可视化的重点是将多维数据在二维或三维空间内显示，这对初步的数据分类理解是有意义的。

14.2.2　科学计算可视化

科学计算可视化是利用计算机图形学和图像处理技术，将工程测量数据、科学计算过程中产生的数据及计算结果转换为图形图像在屏幕上显示出来，并进行交互处理的理论、方法和技术。

科学计算可视化技术主要有两个难点：一是分类，研究如何判断出可视化对象的类别；二是绘制，研究如何将可视化对象真实、高效地显示在屏幕上，使得用户可交互式查看。

14.2.3　信息可视化

信息可视化技术是指利用计算机支撑的、交互的、对抽象数据的可视表示，来增强人们对这些抽象信息的认知。信息可视化是将非空间数据的信息对象的特征值抽取、转移、映射、高度抽象与整合，用图形、图像、动画等方式表示信息对象内容特征和语义的过程。信息对象包括文本、图像、视频和语音等类型，它们的可视化是分别采用不同模型方法来实现。

信息可视化方法根据不同的分类标准进而可分为不同类别，通常按照信息资源本身的特征可将其划分为七类。

（1）一维信息可视化。一维信息是简单的线性信息，如文本，或者是一列数字。

（2）二维信息可视化。二维信息是指包括两个主要的信息。例如宽度和高度可以描述事物的大小，事物在 x 轴和 y 轴的位置表示了它在空间的定位，城市地图和建筑平面图都属于二维信息可视化。

（3）三维信息可视化。三维信息通过引入体积的概念，超越了二维信息。计算机科学计算可视化都是三维信息可视化，因为科学计算可视化的主要目的就是表示现实的三维物体。

（4）多维信息可视化。多维信息是指在信息可视化环境中的那些具有超过三个属性的信息。在可视化中，这些属性的重要性是不言而喻的。

（5）层次信息可视化。抽象信息之间的一种最普遍关系就是层次关系，如文档管理、图书

分类等。对于大型的层次信息结构用可视化技术来表示，可以更加简明和直观。

（6）文档（文本）信息可视化。在如今的信息社会中，各种文档信息堆积如山，可视化技术可以帮助我们快捷地从文档信息中获取我们所需要的内容和知识。文档信息可视化可以分为两类：一类是对单个文档本身的可视化；另一类是对大型文档集合的可视化。

（7）网络信息可视化。目前，Web 的信息已分布在遍及世界各地的数以万计的网站上，网站通过文档之间的超链接彼此交织在一起。网络信息可视化可以帮助人们理解信息空间的结构、快速发现所需信息、有效防止信息迷途。

14.2.4 知识可视化

知识可视化是在科学计算可视化、数据可视化、信息可视化基础上发展起来的新兴研究领域，是所有可以用来建构和传达复杂知识的图解手段。

知识可视化应用视觉表征手段，促进群体知识的传播和创新，它的目标是传输见解、经验、态度、价值观、期望、观点、意见和预测等，并以这种方式帮助他人正确地重构、记忆和应用这些知识。

14.3 多维数据可视化

多维数据可视化是数据可视化的主要内容，它力图在二维或三维空间中展示多属性数据特征，尽量反映数据的各属性信息。

利用多维数据可视化：

- 能够较为容易发现数据变化趋势，如数据的暴涨暴跌等；
- 能够较为容易找出数据异常点；
- 能够较为容易识别数据边缘点，如最大值、最小值、边界数据、新旧数据等；
- 能够较为容易显示数据分类和分簇，并发现不同类数据的特征；
- 能够较为容易地在屏幕上显示更多数据点；
- 能够较为容易地提供丰富的人机交互功能，帮助用户准确地找到特定的数据，并实现对数据的选择、缩放、过滤等基本功能。

对于一个高维的观察对象，若需要用二维图中的多个点球，则称为二维多点表示。典型的二维多点表示方法有雷达图、平行坐标、脸谱图、三角多项等。它能直观地反映同一观察对象中各变量之间的关系，因此适用于对观察对象进行特征提取，是多维数据矩阵的行向量表示，后期可利用基元分类法处理。

二维单点则是表示将观察对象中的全部或部分变量映射为二维图中的一个点。该类表示方法可以在同一幅多元图中显示多个观察对象，从而发现观察对象之间的关系，它适用于数据的特征选择、聚类和分类。典型的有极坐标映射、散点图、星座图等。

图形有助于对所研究的数据进行直观了解。如何将多维数据用平面图来表示，从而显示它的规律一直是人们关注的问题。从 20 世纪 70 年代以来，发展了大量多维数据的图形表示方法。

可视化技术主要有以下几类：面向像素技术、几何映射技术、基于图标的技术、分层可视化技术、基于图表的可视化技术和混合可视化技术等，如图 14.1 所示。

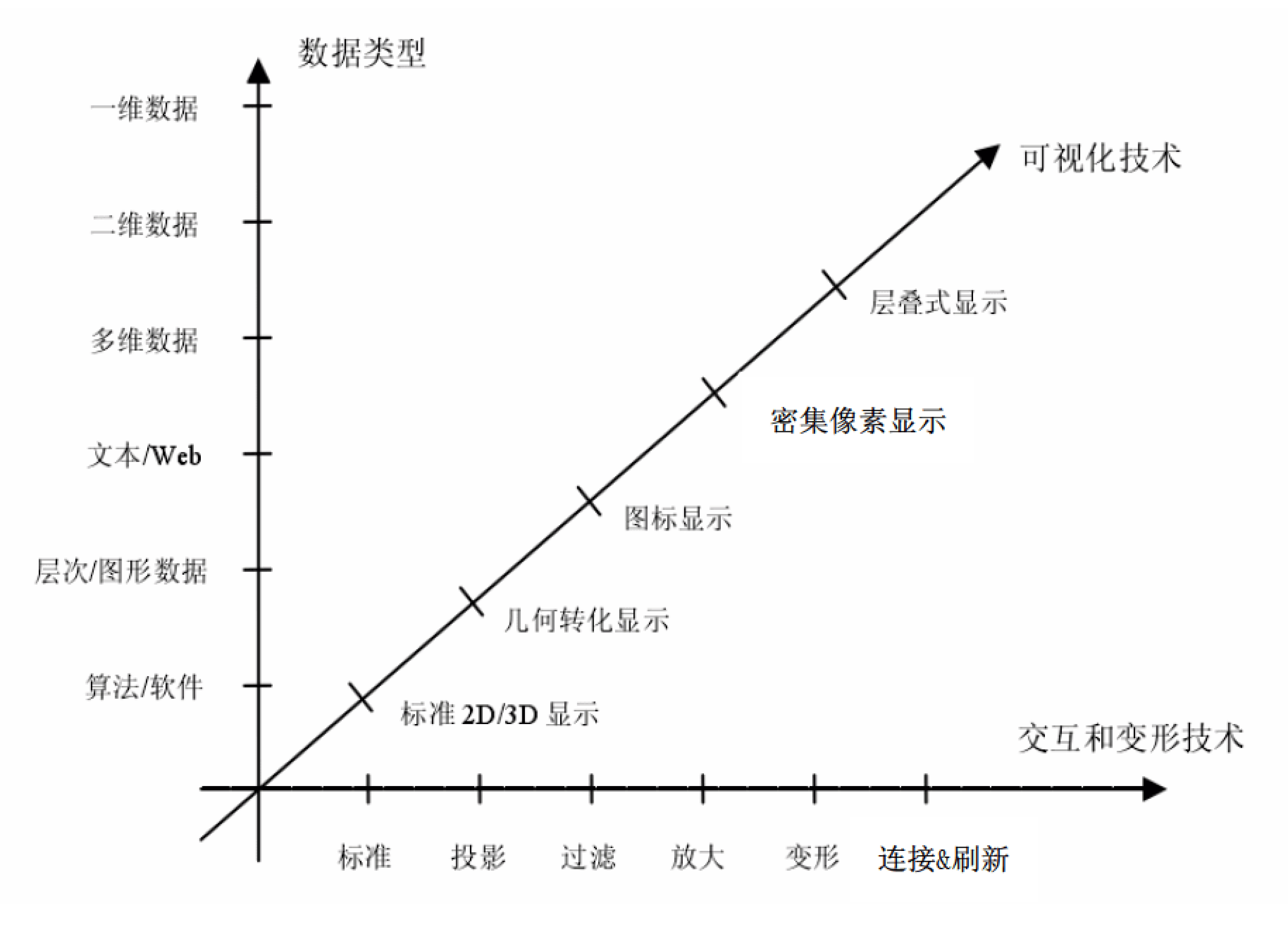


图 14.1　可视化技术

（1）面向像素技术(Pixel－OrientedTechniques)。其基本思想是把每个数据值映射为一个颜色像素。由于一个像素代表一个数据值，所以这种技术可以同时可视化大量数据。面向像素技术又分为独立于查询的可视化技术和基于查询的可视化技术。

（2）几何映射技术（Geomettric Projection Techniques）。几何映射技术的目标在于发现多维数据集的有趣映射，主要包括平行坐标、散点图矩阵、投影追踪等技术，其中较常见的是平行坐标技术。

（3）基于图标的技术（Icon-Based Techniques）。其主要思想是将每个多维数据映射为一个图标，通过观察这些图标组成的图形来发现知识。所用图标种类很多，包括脸形图标、彩色方格矩阵、条形人等。目前最适合可视化大量数据的图标显示技术是条形人技术。

（4）分层可视化技术（Hierarchical Techniques）。把数据分成不同的层次，并在不同的层次上显示。主要用来可视化多变量函数。这类技术包括 N-Vision、The Dimension stacking、Treemaps 等技术。

（5）基于图表的可视化技术（Graph-Based Techniques）。其主要思想是用特定的布局算法、查询语言以及抽象技术来有效地把数据显示成一个大的图表，从生成的图表中发现知识。这类技术包括 Hy+、Margritte 和 seeNet 等技术。

（6）混合可视化技术（Hybrid Techniques）。混合可视化技术是其他一些技术的混合，目的是使各种可视化技术互相补充，更有效地用图形来表现数据。

14.3.1　平行坐标表示法

轮廓图又称轮廓图。它将 m 维欧氏空间的点 x_i（$x_{i1},x_{i2},\cdots,x_{im}$）、线及平面映射到二维平面上的一条曲线，具体步骤如下。

（1）作笛卡儿坐标系，横坐标取个 m 点，以表示 m 个变量。

（2）对给定的一个样本（或观察值），其 m 个点的纵坐标（即高度）与变量取值成正比。

（3）连接 m 个点得一折线，即为该样本的一条轮廓线。

（4）对于具有 n 个样本的数据集，重复以上步骤，可画出 n 条折线，构成整个数据集的轮廓图。对于不同的样本，可以用不同的颜色、线条类型等加以区分。

轮廓图中每个变量都被一致对待，便于使用者可以通过观察多维数据之间联系进行数据挖掘。它还可以作为其他方法的预处理。图 14.2 即为某环境质量监测数据的轮廓图。

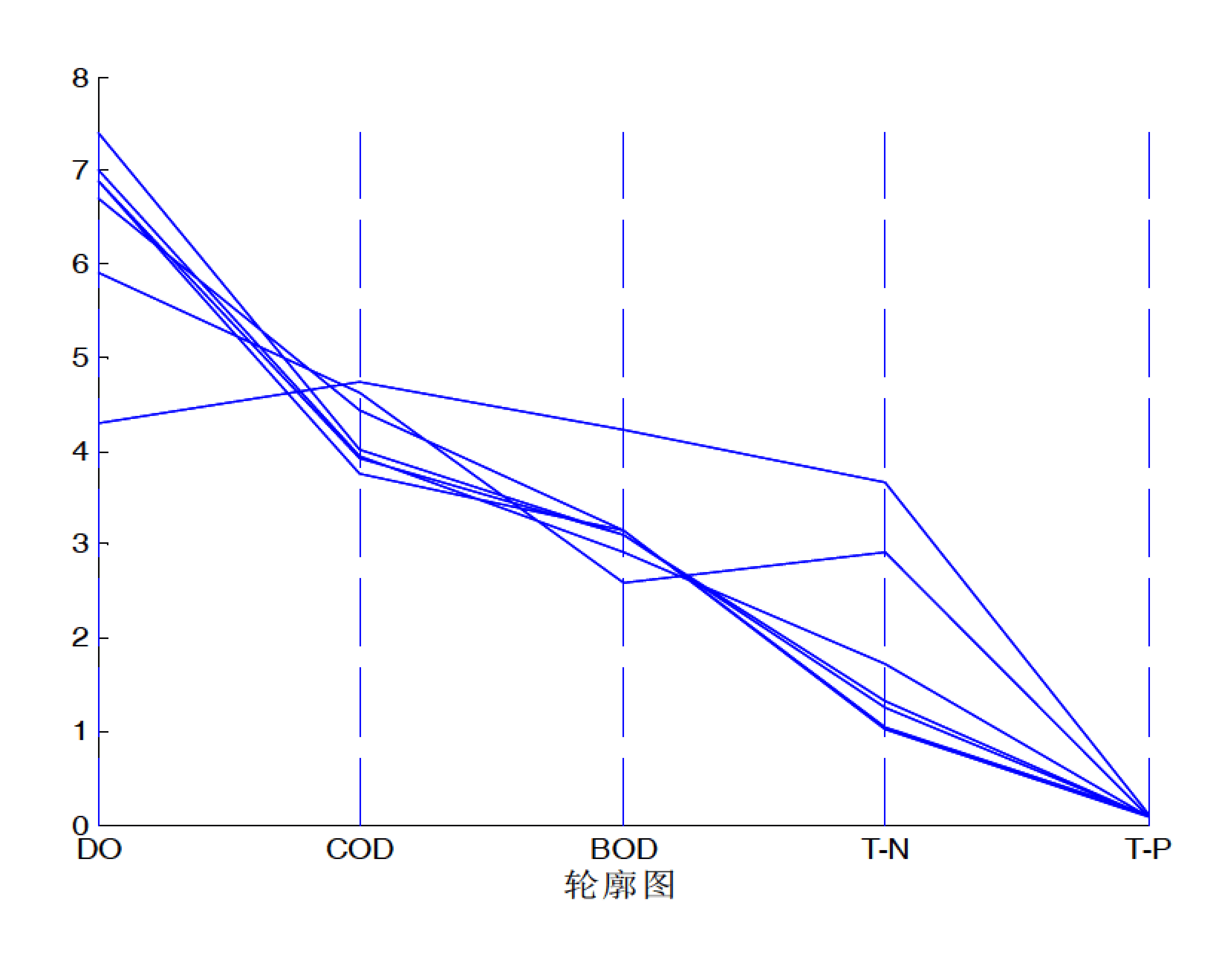


图 14.2 轮廓图

轮廓图的优点是将多维数据用二维的坐标图简单地表示出来，从而达到降维的效果。但是当维数增加时，即所观察的变量增加，映射到平行坐标上表现为平行坐标轴的增加，而随着轴数的增加必然导致轴间距离过于接近，使得图形凌乱，有碍于有用信息的发现，并且坐标轴刻度虽然也表示变量相互间的关系，但是容易造成混淆，数据点连接也可能出现错误。

14.3.2 雷达图

雷达图又称蜘蛛网图，是一种能对多变量数据进行综合分析的形象直观的图形表示方法。由于它有多个坐标轴，可以在二维平面上表示多维数据，因此利用雷达图可以很方便地研究各样本点之间的关系。

绘制雷达图的具体步骤如下。

（1）设原始数据共有 n 个变量，先画一个圆，由 n 个点把圆周等分成 n 个部分。

（2）将圆心和 n 个点连接起来，就可以得到 n 个辐射状的半径，这 n 个半径就作为 n 个变量的坐标轴。这里的坐标轴只有正半轴，因此只能表示非负数据，如要表示负数据，则需进行适当的变换。

（3）为划分刻度方便，在标记坐标轴前需要对原始数据进行归一化处理，然后对归一化后的数据 y_i 用下式作非线性变换

$$f_i = \frac{2}{\pi}\arctan(y_i) + 1$$

通过该变换将无限区间（$-\infty$，$+\infty$）变换到有限区间[0，2]；并使得在均值附近具有良好线性，而偏离均值越远的压缩性越强。

将 n 维数据的各个维规范化的数值刻在对应的坐标轴上，依次连接起来得到一个 n 边形，即得到用平面表示的 n 维数据的雷达图。图 14.3 即为某区域土壤重金属含量的雷达图。

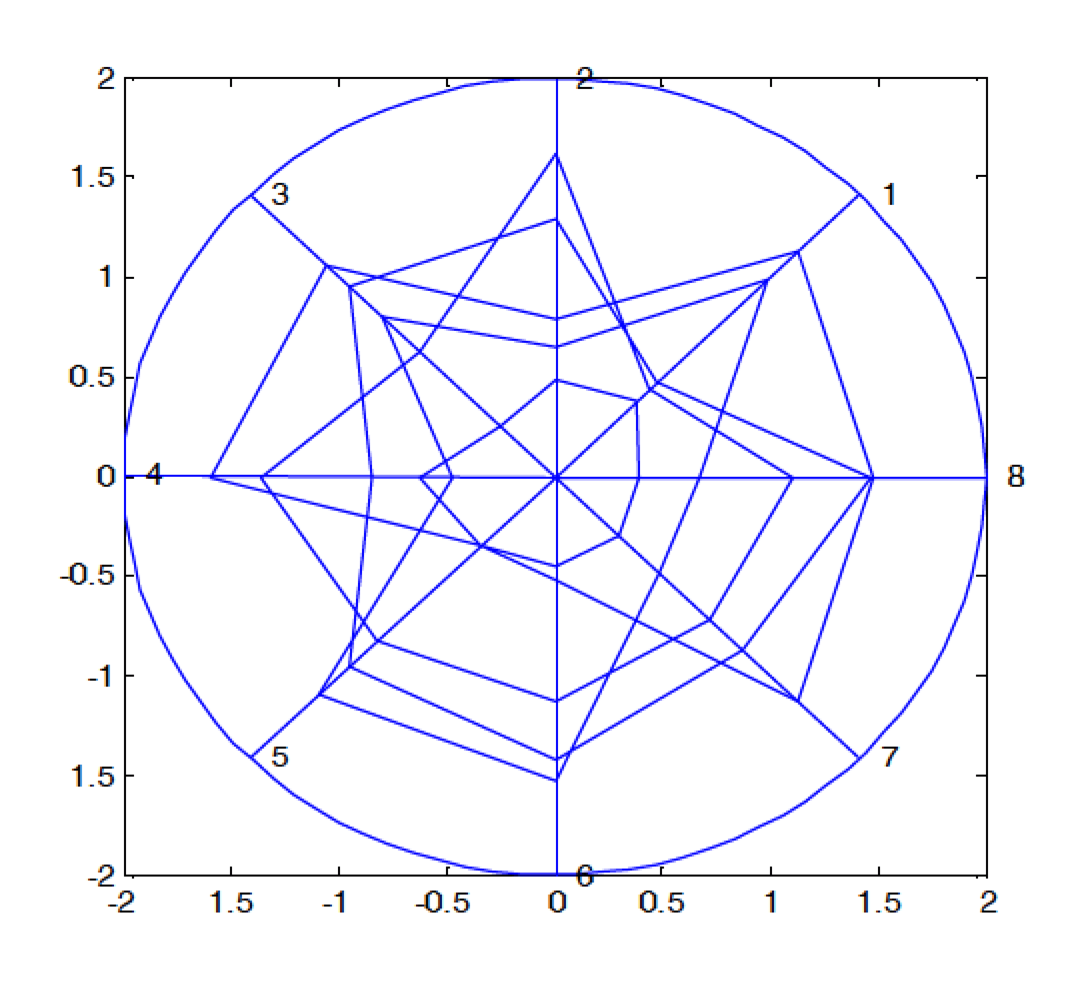


图 14.3　重金属含量雷达图

当要分析的多维数据的个数较少时，可以在同一个雷达图中将它们表示出来；当维数较大时，为使图形清晰，每张图形可以只画少数几个样本数据，甚至每张图形只画一个样本值；或者根据数据的相关性将它们分组，同一组的用同一个雷达图表示，其中不同的多维数据可用不同的颜色的多边形来区别。同时，为了获得更好的效果，在雷达图中适当分配变量的坐标轴，并选取合适的尺度是十分重要的。例如，把要进行对比的指标分别放在其坐标轴左和右或正上方和正下方，以便根据图形偏左、偏右或偏上、偏下进行对比和分析。

如果各参数的权重不一样，则可以根据变量权重的大小分配角度。权重系数或者由其他方法确定，或根据下式求得

$$r_i = \left(\frac{x_i}{x_{i\max}}\right)^2$$

式中：x_i 为第 i 个变量，$x_{i\max}$ 为它的最大值。

雷达图表示方法的主要特点是直观，它能将多维数据映射到二维图形中，可以形象地得到样本数据的状况，并可以对数据得出初步的判断。

14.3.3　树形图

雷达图中，变量的次序是任意的，有时候变量的安排使图形显得茫然，不利于从整体上比较

和评估数据变化的规律性。树形图可以克服这个缺陷。

树形图是用一棵树来表达多个变量，树上每一个末树枝对应一个变量，这棵树的分叉的位置与角度，即变量的次序是根据层次聚类的原则确定的，主干树取决于分枝聚类时的主导变量，而分枝按相关程度依次从高到低排列。末枝的长度表示变量的观察值，分枝的长度是其上末枝长度的平均值，分叉的角度等价于两变量间相关系数 r_{ij} 的夹角余弦。令 θ_{ij} 表示变量 x_i 和变量 x_j 之间的夹角，则

$$\theta_{ij} = \arccos r_{ij}$$

即相关性强则夹角小，相关性弱则夹角大。如此依相关程度层层聚类，直至最后的树枝而形成一棵完整的树。图 14.4 即为树形图，图中的大枝是由分量 x_1 和 x_8 决定：

$$x_{12} = \frac{x_1 + x_2}{2},\quad x_{34} = \frac{x_3 + x_4}{2},\quad x_0 = \frac{x_1 + x_2 + x_3 + x_4}{2}$$

$$\theta_{12} = \arccos r_{12}, \theta_{34} = \arccos r_{34}, \theta_0 = \arccos r_{18}$$

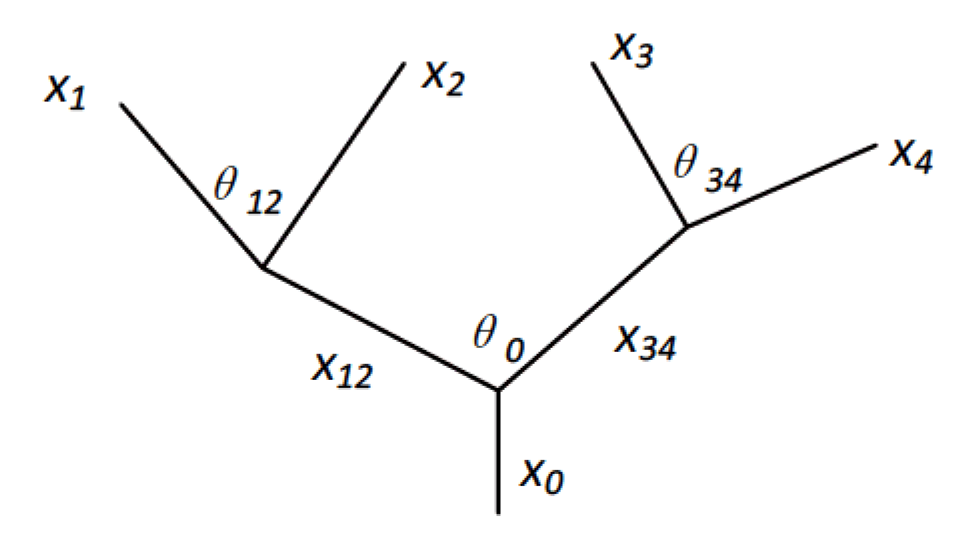


图 14.4 树形图

画树形图前，首先对数据进行层次聚类以得到聚类树，由聚类树画出多元树形图是很容易的。

14.3.4 三角多项式图

三角多项式图又称调和曲线图，它是以三角多项式作图来实现的。通过三角多项式把多维空间中的一个样品用二维平面中的一条曲线来表示，并希望这条曲线能够保留原数据的全部信息。它既可以应用于数据的分类和聚类，也可以用来发现异常点。

绘制三角多项式曲线的具体步骤如下。

设有 p 维数据

$$x = (x_1, x_2, \cdots, x_p)$$

则其对应的平面曲线为

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin t + x_3 \cos t + x_4 \sin 2t + x_5 \cos 2t + \cdots, -\pi \leqslant t \leqslant \pi$$

当 t 在区间$[-\pi,\pi]$上变化，其轨迹是一条曲线，若多个数据按照同样办法作图，就会对应多条曲线在平面上，这就构成了调和曲线图，如图 14.5 所示。

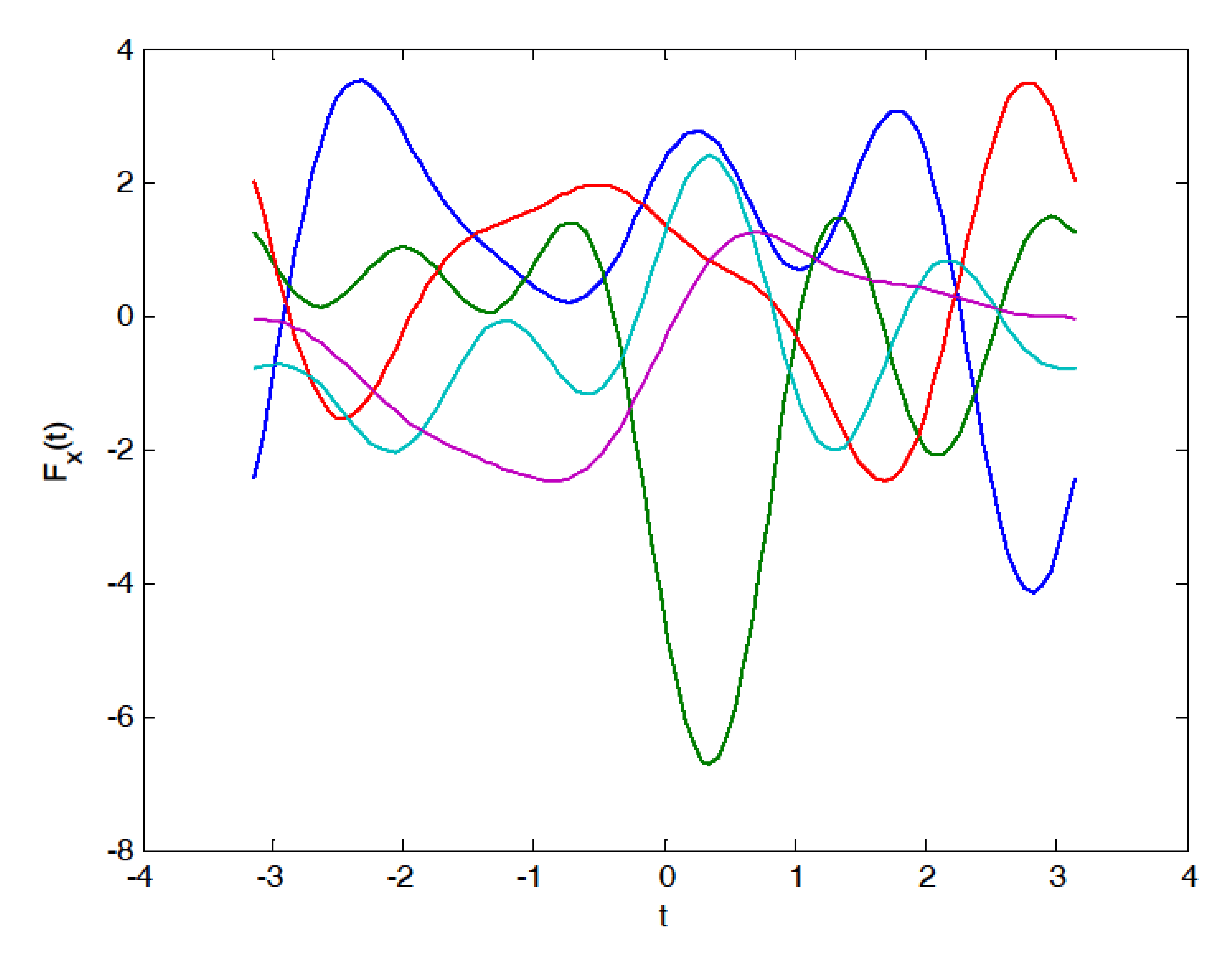


图 14.5　三角多项式图

14.3.5　散点图

散点图是将多维数据以平面或空间中的点来表示，最常用的是二维数据在笛卡儿坐标系内表示的情况，称为直角散点图或 XY 散点图。有时为了更好地描述多维数据的变化趋势，用直线或平滑曲线将各数据点连接起来，而成为折线图和平滑线散点图。XY 散点图能描述的是包含两个变量的二维数据，在使用这种方法描述高于二维的多维数据时，常用散点图矩阵来表示。

另一类散点图称为三角散点统计图或 XYZ 散点统计图，它用等边三角形的三条高为坐标构成的“三角坐标系”内描述 3 个变量，每一散点代表 3 个对应的变量值。该方法常用来描述一类称为概率单纯形的数据，这类数据所包含的若干个变量指标之和为一个常数。

1. 直角散点图

直角散点图实际上就是多维数据在多维空间中的坐标点表示，各维坐标对应多维数据中的各变量值。实际上应用最多的是平面直角散点图，即 XY 散点图。

二维数据的平面散点图表示方法非常简单，实际上就是将二维数据（x,y）在笛卡儿坐标中描点表示。

2. 散点图矩阵

平面直角散点图所能描述的是包含两个变量的二维数据，对于高于二维的多维数据，常用散点图矩阵来表示。散点图矩阵可以看作一个大的图形方阵，其每一个非主对角元素的位置是对应行的变量与对应列的变量的散点图，而主对角元素的位置上是各变量名，这样借助于散点图矩阵能清楚地看到所研究的多个变量两两间的关系。

散点图矩阵的各元素位置散点作图方法和两变量散点图完全相同。

3. 三角形散点图

三角形散点图表示多维数据仍以平面或空间内的一点来表示，应用较多的是三维概率单纯形的数据在平面上的表示，即 XYZ 散点统计图。

三角形散点图中的正三角形的 3 条高分别表示 3 个变量的坐标轴，高的底为 0，顶点为 1（即 100%）。很明显，3 条坐标轴交于坐标为（1/3，1/3，1/3）的一点，同时三角形内任意一点 A 到三边的距离之和为常数 1，这样任何三维概率单纯形的数据均可用等边三角形内的一点表示。

14.3.6 星座图

星座图就是将 n 个样品点在一个半圆内表示，一个样品用一颗星表示，同类的样品组成一个星座，不同类的样品组成不同的星座，所以形象地比喻为星座图。

星座图是一种非常直观的方法，在对多个指标的数据在不同的权重下进行汇总时，具有既能体现统计数据的统计结果，还能反映数据的均衡性的优点，因此，使用极其方便。根据样本点的位置可以直观地对各样本点之间的相关性进行分析，利用星座图还可以方便地对样本点进行分类，在星座图上比较靠近的样本点比较相似，可以分为一类，相距较远的点相应样本的差异性较大。

绘制星座图的具体步骤如下。

（1）为消除量纲的影响，将数据作线性变换，使更换后的数据落在某一线性范围内。常用的线性变换方法为极差标准化，使变换后的数据落在$[0,\pi]$闭区间内，其变换公式为

$$y_{ij}=\frac{x_{ij}-x_{\min,j}}{R_j}\cdot\pi$$

式中：R_j 为数据矩阵每列的极差。

（2）适当选取一组路径权重$\{\omega_j\}$，使满足

$$\sum_{j=1}^{p}\omega_j=1$$

$$\omega_j\geqslant 0,\quad j=1,2\cdots,3,\cdots,p$$

重要变量相应的权重可以取得大一点，但一般情况下可以取等权，即

$$\omega_1=\omega_2=\cdots=\omega_p=\frac{1}{p}$$

（3）画一个半径为 1 的上半圆及半圆底边的直径，使每个样本对应半圆内的一个点，称为星，这些星就落在这个半圆内。设有模式 X_1，首先以为半圆心为圆心，ω_1 为半径，画上一半圆，在圆周上对应弧度为 y_{11} 的点为 O_1，然后再以 O_1 为圆心，以 ω_2 为半径画一个半圆，在圆周上对应弧度为 y_{12} 的点为 O_2，以此类推，直至 O_p 为止。O_p 即为 X_1 与对应的星座的位置。由 O 点通过上述作图步骤，到达星的路线称作该星座的路径，由以上可得出与任一模式 $X\alpha$ 对应的星座位置坐标为

$$(\sum_{j=1}^{p}\omega_j\cos y_{\alpha j},\sum_{j=1}^{p}\omega_j\sin y_{\alpha j})$$

通过星的位置和路径就可以全面刻画该样本的特征，图 14.6 即为星座图。根据星座图上点的位置及路径判断各样本间的接近程度，进而可以对样本点进行归类分析。在实际工作中，人们往往去掉样本点的路径部分而仅保留其在星座上的位置，并根据各点位置的接近程度分析样本点间的接近程度。

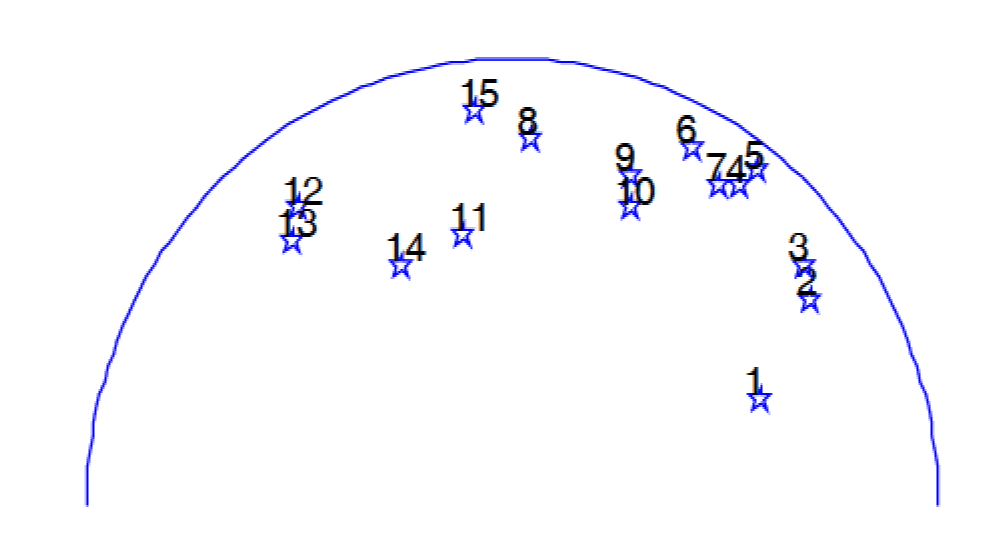
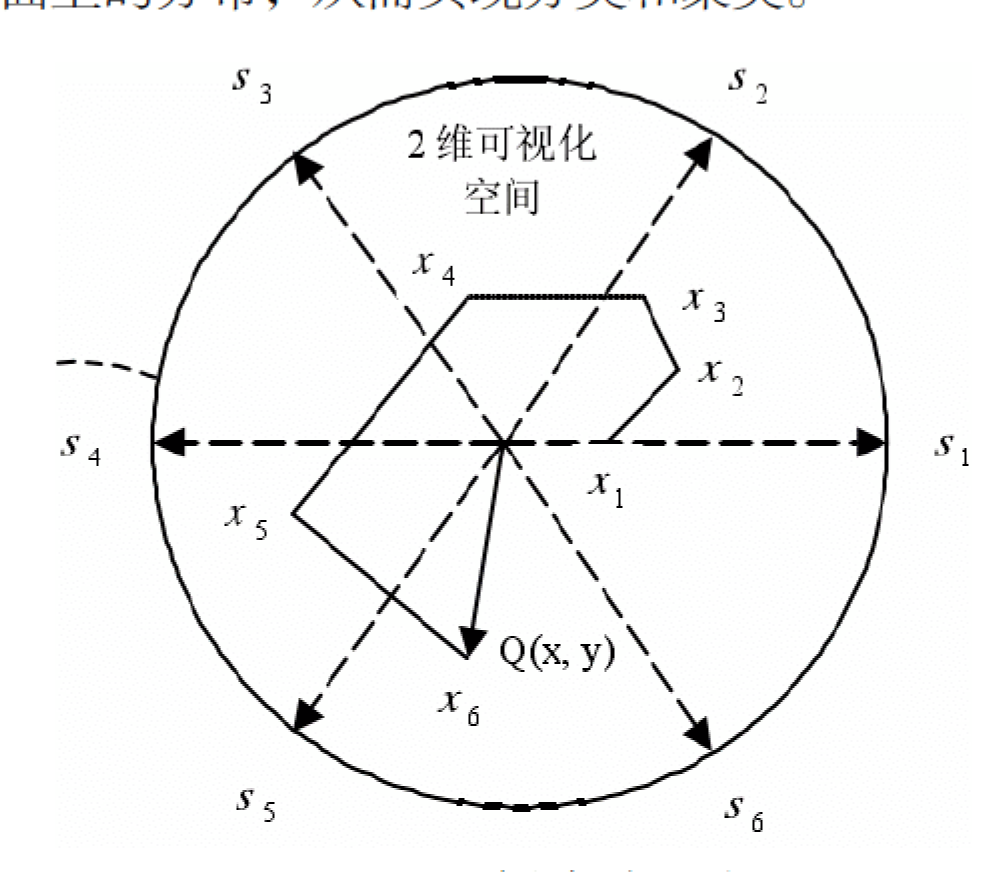


图 14.6　多维数据的星座图

当样本数较大时，数据在一个半圆内显得比较“拥挤”，且易造成“殊途同归”的现象，给分类带来了一定的困难。此时，可以通过适当“拉开”样本距离，即将数据扩充到半径为 1 的整个圆内（2π 区间），就可充分利用原始数据的信息，各样本间的区别与联系将更加清楚，为合理分类提供了方便。

与星座图相似的是星型坐标表示法，如图 14.7 所示。它的基本思想是在一个二维平面上排列一系列的坐标轴，这些坐标轴并不是正交的，每一个坐标轴都对应一个数据维，n 维数据属性以坐标轴的形式映射到二维平面上，n 维数据空间中的点被表示成二维平面上的一个点。在二维平面的圆上排列了许多坐标轴，轴间角度相等，原点是圆的中心。轴的长度与数据值成比例，最小值映射到圆点，最大值映射到轴的另一端，此轴段即为该轴的单位向量。通过调整轴长和角度，可以调整数据集在二维平面上的分布，从而实现分类和聚类。



14.7　星型坐标表示法

通过改变坐标轴长度，可以提高或降低某一维或多维数据对可视化结果的影响；改变坐标轴的方向，可以提高或降低相应维数据与其他维数据的关联。旋转同样解决图像重叠问题，既可以将重叠的点分离，也可以将重叠在一个区域的不同类簇分开。

另外，还可以选择图中的单个点或某个范围来标记数据点，数据点将被标记成所选择的颜色。通过标记数据，可以方便地观察数据子集的变化情况。

14.3.7 基于像素的高维数据的可视化

面向像素技术的高维数据可视化技术的基本思想是将 n 维对象映射成一个圆，并将圆划分成 n 段代表不同的属性。每个属性值映射成一个颜色像素，并用分隔子窗口代表属于不同维度的属性值，像素的颜色由 HIS 颜色范围确定，HIS 颜色范围是对 HSV 颜色模式进行轻微修改而成。在每个子窗口中，相同记录的属性值被标记在相同的相对位置上。

图 14.8 显示了通过对白、灰和黑三种颜色的离散化，将 30 000 条包含 16 个描述工业部件周线和可达性属性维的记录可视化，可达性属性维的描述清楚给出了总体的聚类结构以及许多由小到大的聚类展示。只有描述序列末尾段的外边界展示了一个大的由代表噪声的白色区域围成的大的聚类。在大的聚类中比较属性级数，很显然，属性维 2~9 表现了一个恒定的值，而其他属性维在倒数第三部分数值上出现不同，此外，与其他属性维相比，属性维 9 的最小值位于大的聚类中，而其最大值位于其他聚类中，集中观察像可达属性维中第三条条纹这样的小聚类，可以看出属性维 5、6、7 在许多突出的形式上不同于其邻接属性维。当选择小的聚类，并通过可达区详细进行可视化时，许多另外的数据特性能够展示出来。

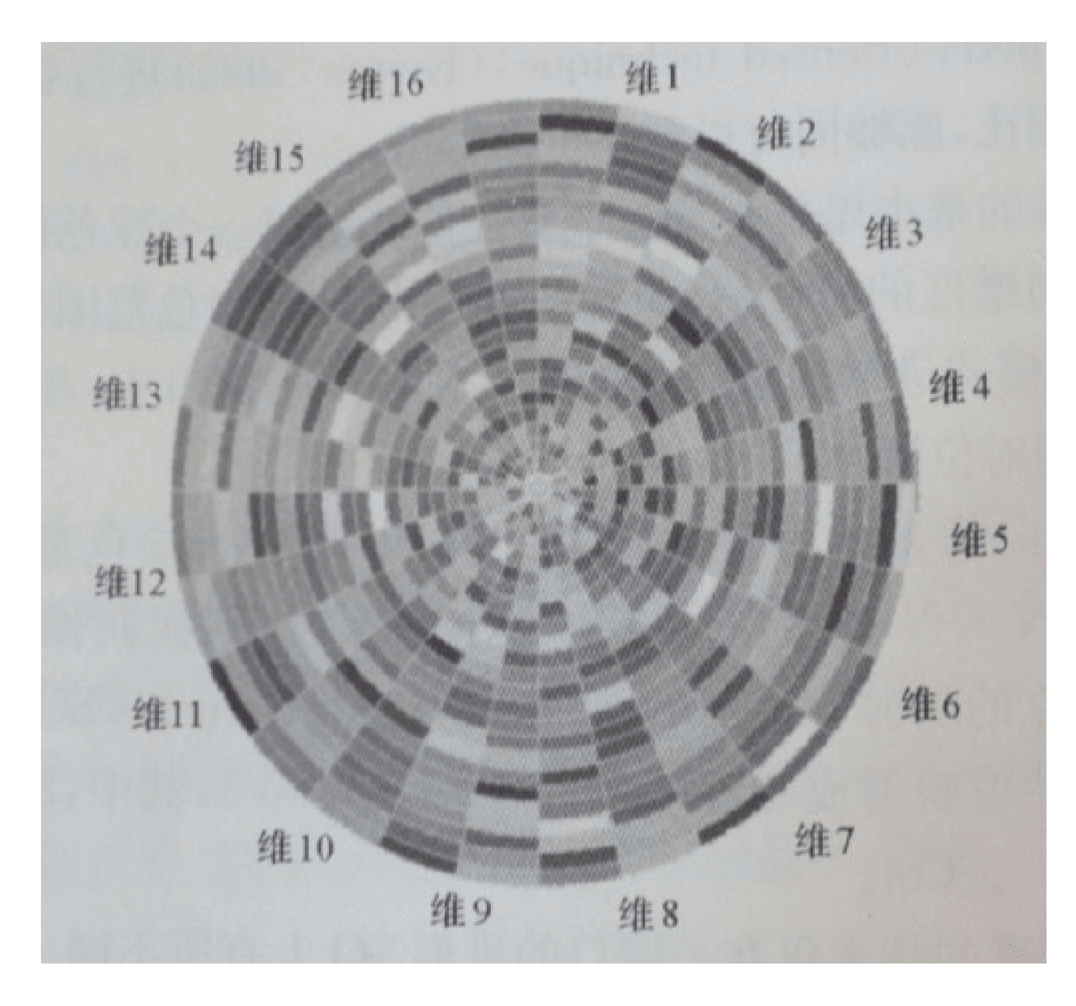


图 14.8　16 维的 30000 个对象的聚类结构

需要注意的是，此技术要求数据至少是三维，另外，用户可以通过改变维的数据在圆内位置以进一步比较数据特性，而且通过色彩控制，将数据点的值映射到像素点的色彩值，一种颜色代表一类数据，当一维数据属性出现在多个分支点情况下就会有多种颜色出现在一个段内，可以清楚地看出分支点的位置。

14.3.8 基于非线性变换的图表示优化

高维数据的图形表示一般都较为复杂，这不利于后续的处理。根据后续处理（如分类）方法的要求，基于非线性函数的优化方法，可以使图表示更符合实际处理要求以及可视化与交互式原则。

在对数据进行非线性描述时，非线性函数 $f(x)$ 选择的基本标准如下。

（1）$f(x)\in[x_{\min},x_{\max}]$，当 $x\in[x_{\min},x_{\max}]$

（2）$x_1\in[x_{\min},x_{\max}]$ 且 $x_2\in[x_{\min},x_{\max}]$ 范围的，当 $x_1\geqslant x_2$ 时，有 $f(x_1)\geqslant f(x_2)$

第一个条件即为值域要求；第二个要求即为单调非减函数要求。满足这两个要求的函数都可以作为转换函数而用于图表示的优化，例如指数函数、多项式函数和分段函数等。

1. 指数函数

根据两个条件的要求，约束后的指数函数为

$$f(x)=\frac{a^x-a^{x_{\min}}}{a^{x_{\max}}-a^{x_{\min}}},a>1$$

如果对其归一化，则有

$$f(x)=\frac{a^x-1}{a-1},a>1$$

2. 多项式函数

多项式函数 $f(x)=\sum_{i=0}^{n}a_i x^i$ 为平滑的连续函数，其二阶导数符号不唯一，经过限定后可以用于多元图的优化。具体做法是取多项式曲线中单调部分，并对其进行缩放与平移，其中最为简单的多项式为

$$f(x)=x^a,a>0$$

当 a<1 时，数据向大数据端汇聚；当 a>1 时，数据向小数据端汇聚；当 a=1 时，数据分布不变。

3. 分段函数

对于某些多元图，可根据其分布的特性采用分段函数处理。

$$f(x)=\begin{cases} f_1(x),x\in[x_{\min},x_1] \\ f_2(x),x\in[x_1,x_2] \\ \vdots \\ f_n(x),x\in[x_{n-1},x_{\max}] \end{cases}$$

式中各子函数均应满足 x 的约束条件。

在分段函数中，“放大镜”函数具有良好的局部放大作用，其表达式为

$$f(x)=\begin{cases} ax+b,x\in[x_1,x_2]\in[x_{\min},x_{\max}] \\ 0,x\notin[x_1,x_2] \end{cases}$$

式中：$a=\frac{x_{\max}-x_{\min}}{x_2-x_1},b=\frac{x_{\min}x_2-x_{\max}x_1}{x_2-x_1}$。

14.3.9 高维数据降维

由于高维数据受到三维物理空间的限制，其图形表示和对数据结构的直观理解比较困难。虽然现在有上述描述的等多种方法可以将多维数据用平面图形来表示，但如果对高维数据进行必要的降维，则对后续的数据处理有极大的帮助。

降维就是在保持原始数据主要特性基础上将高维空间映射到低维空间。作为分析和研究高维数据的重要手段，降维问题具有重要的理论与应用价值，正引起人们越来越多的关注。下面即为

几种常用的高维数据降维方法。

1. 主成分分析

主成分分析是在保证数据信息损失最小的前提下，经线性变换和舍弃一小部分信息，以少数新的综合指标取代原始的多维指标（变量）来反映多维变量所提供的信息。得到的新的综合变量称作主成分。

该方法适用于变量之间存在较强相关性的数据，一般认为当原始数据大部分变量的相关系数都小于 0.3 时，运用主成分分析不会取得很好的效果。

2. 因子分析

因子分析通过对数据矩阵进行特征分析、旋转变换等操作，可以获得数据的相关信息。在因子分析中是由几个潜在的但不能观察的随机变量（即因子）去描述许多变量间的协方差关系，根据相关性的大小把变量分组，使得同组内的变量直接的相关性较高，而不同组的变量相关性较低。

可以把因子分析当成主成分分析的一个扩充。两者都可以看成在力图逼近协方差矩阵。主成分分析中的主成分个数与变量个数 p 相同，它将一组具有相关关系的变量变换为一组互不相关的变量，实际应用时，一般只选择前 m 个（$m<p$）主成分。而因子分析的目的是要用尽可能少的公因子，以便构造一个结构简单的因子模式，将原始变量表示为公因子和特殊因子的线性组合，用假设的公因子来解释相关矩阵的内部依赖关系。很明显基于因子分析模式的降维方法更为精细。

3. 基于特征选取思想的降维方法

由于数据的处理实际上就是对其特征的分析，而各种多维数据的特征可以分为物理的、数学的和结构的特征，因此可以通过特征选择和特征实现降维。特征选择是从一组特征中舍弃一些原始特征而挑选出一些最有效的特征，以达到降低特征空间维数的目的；特征是通过映射（或变换）的方法将高维数据在低维空间来表示样本的过程。因此特征提取和特征选择的基本任务是从众多特征中找出那些最有代表性、最有效的特征，并舍弃一些冗余变量，进行有效分类。图 14.9 即是对于小高维数据的基于特征选取思想降维方法的示意图。

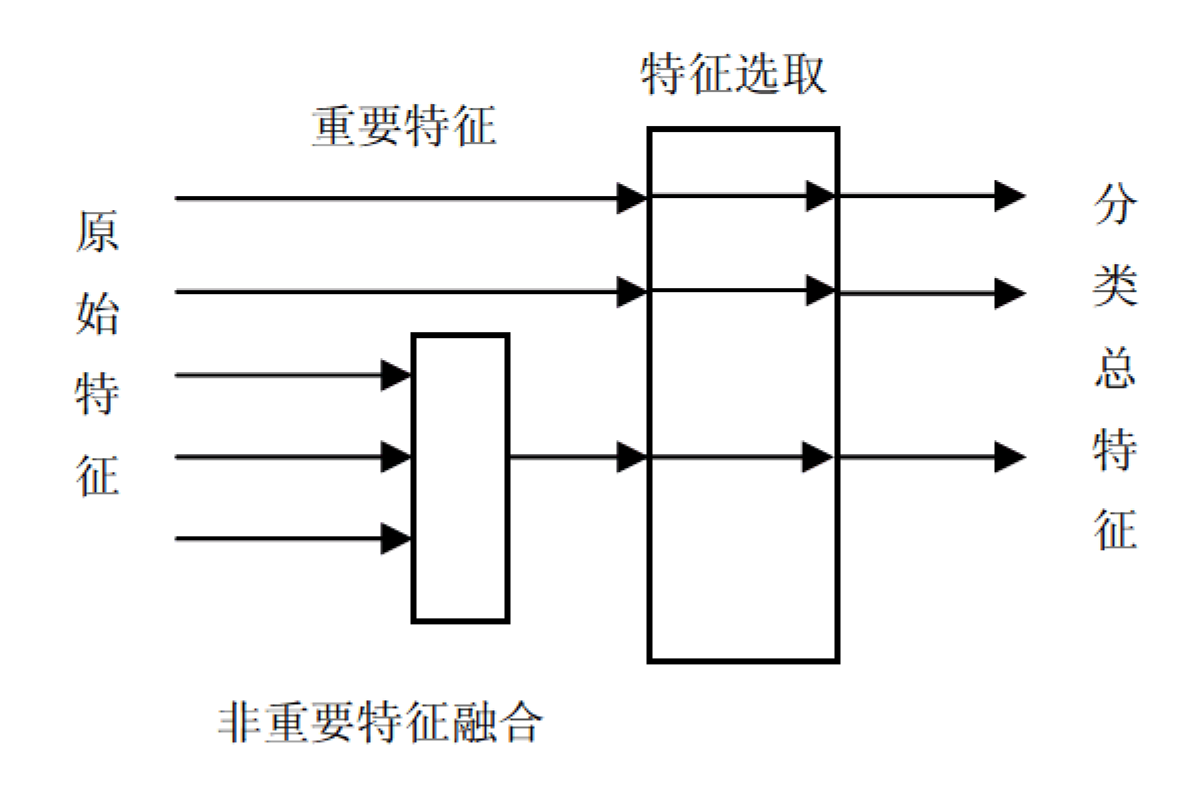


图 14.9　特征选取降维方法示意图

而对于中、高维数据，因为有很多主要的特征，也有很多无关的特征，为此需要采取多级信

息选取的方法进行降维。首先对高维数据进行分段处理，然后对分好的各段进行特征选取，得到代表此段数据的多个特征，其次将所有段的特征组合作为下一级处理的输入数据，直至最后得到能表达高维数据的总特征或模式表达结果。图 14.10 即为该方法的示意图。

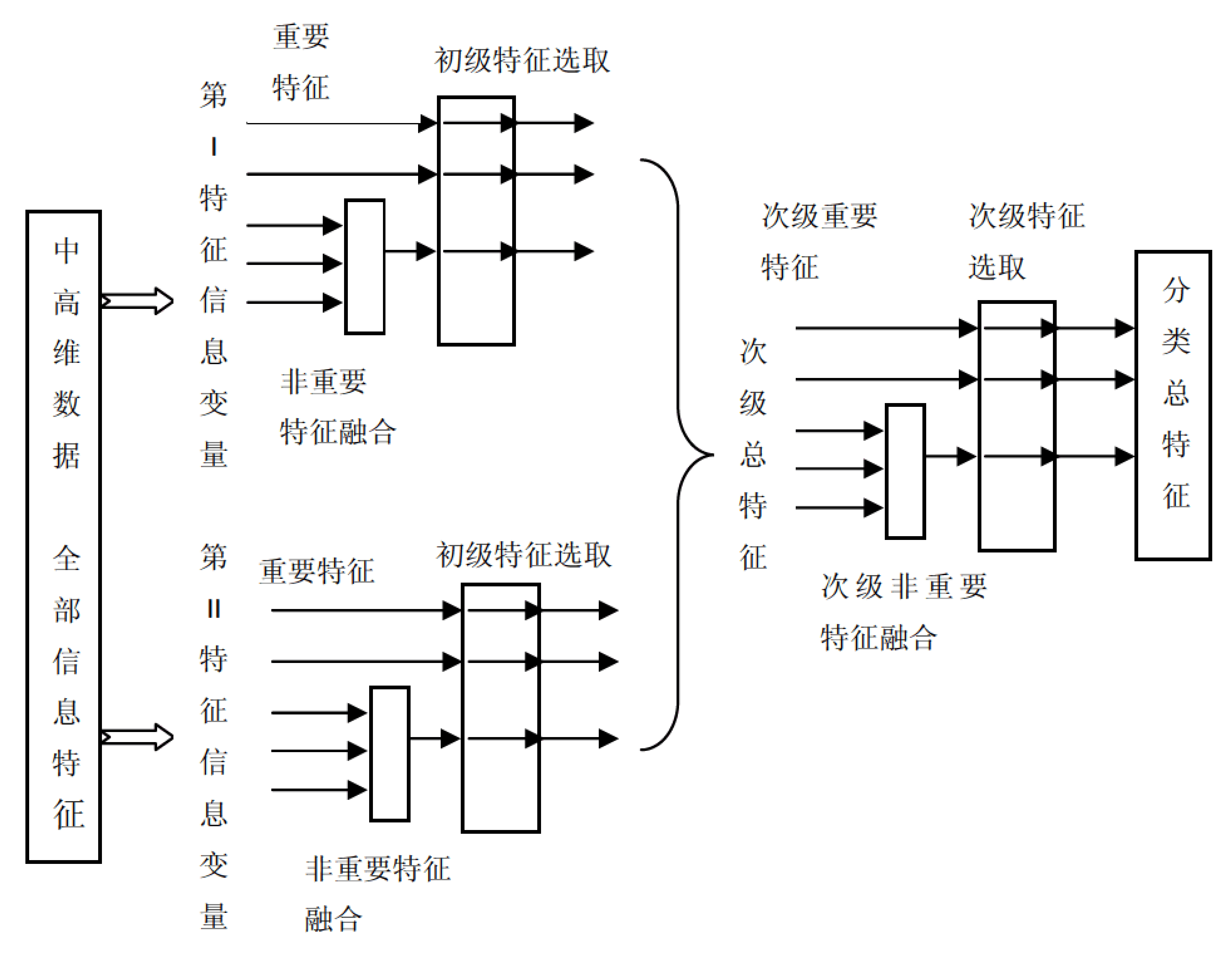


图 14.10　中、高维数据降维处理示意图

14.4　图形的特征分析

多元图表示是数据探索性分析的重要手段，可以获取对数据结构特征的认识指导分类算法选择与参数确定等。

14.4.1　平行坐标下的聚簇分析

用平行坐标可把多维空间的数据集映射到二维平面，为数据特征的分析提供了方便。在平行坐标系下，每一坐标轴实际反映的是数据在该坐标轴上的一维投影，因此在此坐标轴上数据的分离及聚簇明显，有助于发现数据的聚类特征，在多维空间数据的聚簇情况则可同平行坐标中各折线的分离及聚簇情况表达出来。图 14.11 表现的即为四维空间中 3 个独立聚簇。由于平行坐标中可以同时看到各维之间的联系，因此可以很清楚地通过各维观察数据的聚簇情况。

当聚类算法产生的结果包含大量数据对象的多个聚簇时，由于大量的交叠使折线密度增加，生成的图形存在大量的重叠，层次不清，使图形难以辨识。此时可采用 M-BIRCH 算法的分层方式对数据聚类结果进行可视化，进而识别出各个聚簇之间的关系，最终达到准确的聚类。该方法将聚类结果可视化设定在不同的层次上，在用户的参与下，过细的划分的情况下对其父节点（上

级节点）可视化，称为层次上卷，而阈值过大的节点显示其节点（下级节点），即层次上钻，使最终可视化结果是不同层次的、不同粒度的簇的显示，从而解决底层无法完全显示的问题。

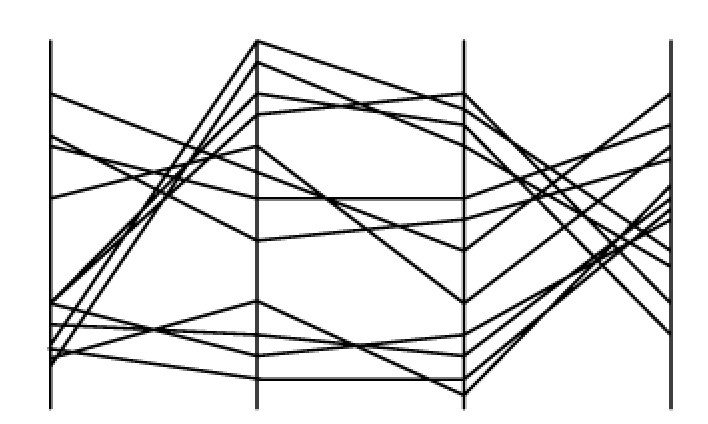


图 14.11　平行坐标的聚簇表示

图 14.12 中的 *a* 表示 M-BIRCH 的逻辑结构框架，*b* 为显示层（可视化显示的基本层），因为 M-BIRCH 采用的数据存储结构为高度平衡树，所以层次可以用直线表示，*c* 为细化层，将处于细化范围内的节点由细化层内的子节点代替进行可视化处理，*d* 表示设定的细化区域，对细化区域包含的显示层进行细化，（e_{min},e_{max}）表示细化范围，对显示层中处于该范围的节点进行下钻，根节点与（e_{min},e_{max}）形成细化区。通过细化范围的左右调节可以修改细化范围，上下调节挖掘数据的上卷和下钻。在分层显示中可以采用一个细化范围，也可以包含多个细化范围。

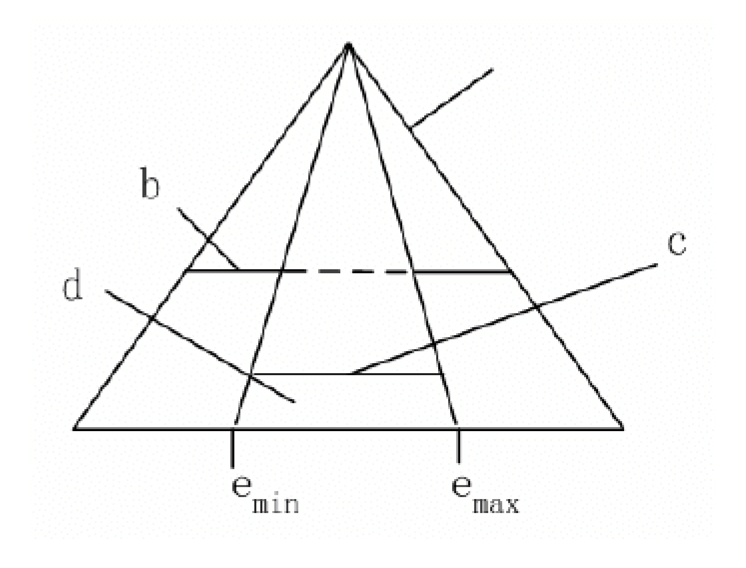


图 14.12　分层显示的概念图

14.4.2　雷达图的图形特征方法

雷达图是二维平面内的一个封装的不规则的多边形,明显的一个结构特征即是由多个三角形组成的多边形，每个三角形都是由相邻变量组成。它的一个明显的视觉特征是多边形的面积和重心。

1. 面积图形特征

对面积的求解，可以采用三角形面积法和扇形面积法，其中基于三角形面积的图如图 14.13 表示。相关符号：面积 *S*、射线 r_i、弧度 ω_i 和维数 *n*。

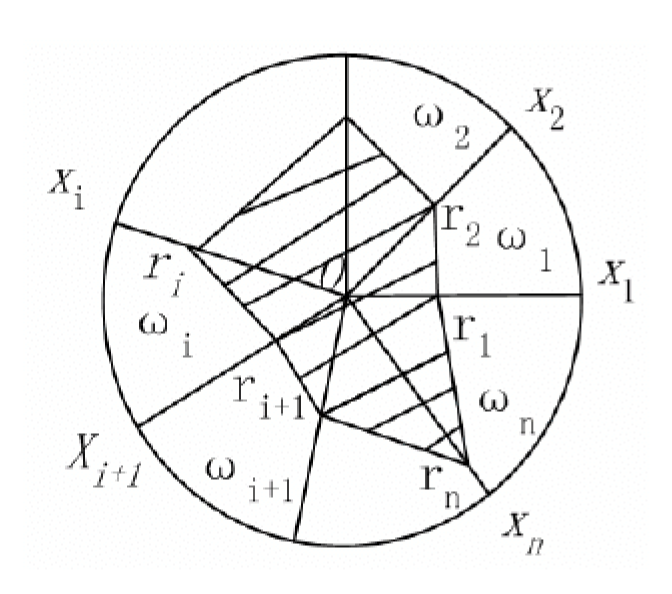


图 14.13　基于三角形面积的图表示原理

单调函数的面积求解方法

$$S = \sum_{i=1}^{n} S_i$$

三角形面积

$$S_i = \frac{1}{2} r_i r_{i+1} \sin \omega_i$$

扇形面积

$$S_i = \omega_i \pi r_i^2$$

2. 重心图形特征

重心是雷达图多边形的另一个信息，如图 14.14 所示。

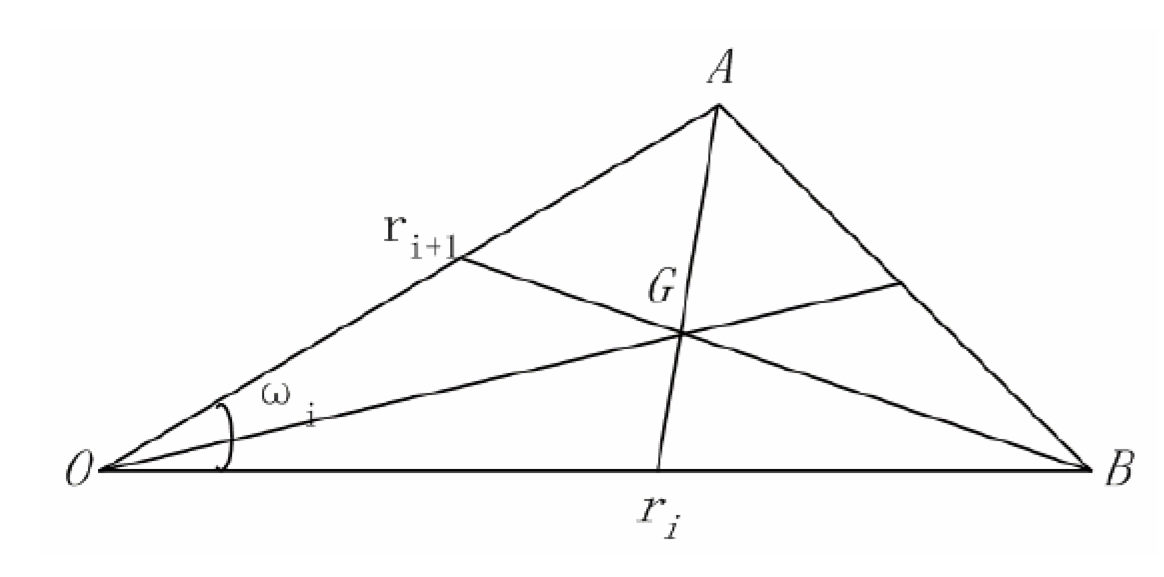


图 14.14　雷达图中一个三角形重心图形特征的示意图

重心的计算公式如下

$$\mathrm{abs}_i = \sqrt{(\frac{r_i}{3}\sin \omega_i)^2 + ((r_i \cos \omega_i) - \frac{r_{i+1}}{2})/3 + \frac{r_{i+1}}{2})^2}$$

$$\mathrm{angle}_i = \arcsin\left(\frac{\frac{1}{3}\sin \omega_i}{\mathrm{abs}}\right) + 2\pi(i-1)/d \qquad i=1,2,\cdots,d$$

其中：abs_i、angle_i 分别为第 i 维变量和第 i+1 维变量组成的三角形的重心的幅值和真实的角度。$\omega_i=2\pi/d$ 为第 i 维变量和第 i+1 维变量间的夹角弧度，可以认为圆角被样本维数 d 平分。

重心计算出来后，就可以将重心的幅值作为雷达图重心图形特征。一个 d 维特征的样本就会产生一个对应的 d 维重心图形特征的样本。

14.4.3　图形特征提取中的特征排序问题

实验表明，多元数据的特征排序不同会导致不同的雷达图，进而会影响到分类性能。所以在实践应用中要注意特征排序的问题，要寻找最优分类所对应的特征排序。

解决特征排序的方法有两种：一种是采用传统的特征选择方法，包括穷举法、单独最优特征组合法、顺序前进法、顺序后退法等；第二种是采用基于全局优化的随机搜索算法如遗传算法等；第三种是利用准则函数寻找最优特征顺序。

穷举法是要把所有可能的特征组合都计算一遍，然后再从中找出最优的特征排序。很明显，当维数较大时，这个算法是无法实现的。这样使得寻找一种可行的算法变得非常重要。但要注意，所有算法原则上仍是穷举法，只不过采用某种搜索技术使计算量可能有所减少。

第一种方法在理论上就不能保证是最优的特征排序；第二种方法如遗传算法在实现上又十分耗时，第三种基于准则函数的方法既能找到较优的特征排序，也能减少时间开销，其计算过程如下：

首先计算每两维间的 Spearman 相关系数，然后找到相关系数最小的两维 h 和 k，接着对这两维中的每一个分别计算与剩余其他维间的 Spearman 相关系数，然后找到相关系数最小的两维 k 和 t，那么，特征排列的头三位是 h、k 和 t，接着计算 t 与剩余其他维的 Spearman 相关系数，然后找到相关系数最小的对应的维，这一维就是特征排列的第 4 位，以此类推，直到考虑所有维。最终得到的特征排序是相邻两维的 Spearman 相关系数从头到尾依次减少。

Spearman 相关系数的计算公式如下

$$\rho_{i(i+1)} = 1 - 6\frac{\sum_{j=1}^{N}(r_{ij} - r_{(i+1)j})^2}{N(N^2-1)}$$

式中：N 为样本的总个数；i、i+1 分别为第 i 维和 i+1 维；j 为样本索引值；r_{ij} 为第 j 样本第 i 维。

14.5 基于多元图的图形分类方法

根据 Penrose 创立的形状差异系数，可以将下式定义为基于面积特征的图形相异度系数

$$z_{ij} = \sqrt{\frac{d}{d-1}(r_{ij}^2 - q_{ij}^2)} \qquad q_{ij}^2 = \frac{1}{d^2}(\sum_{n=1}^{d} s_{ni} - \sum_{n=1}^{d} s_{nj})$$

式中：r_{ij}^2 为样本点 i 和 j 的欧氏距离，s_{ni} 为样本点 i 的雷达图上的第 n 维围成的扇形的面积值，或者是第 n 维和下一维围成的三角形的面积值，即第 n 维面积特征。

类似地，可以定义基于重心特征的图形相异度系数

$$z_{ij} = \sqrt{\frac{d}{d-1}(r_{ij}^2 - q_{ij}^2)} \qquad q_{ij}^2 = \frac{1}{d^2}(\sum_{n=1}^{d} g_{ni} - \sum_{n=1}^{d} g_{nj})$$

式中：g_{ni} 为样本点 i 的雷达图上的第 n 维围成的三角形的重心坐标幅值，即第 n 维重心特征，其余符号意义与前相同。

14.5.1 单原型图形分类器

定义了图形相异度系数后，便可以按照模板匹配思想用基于图形相异度的分类器对图形进行分类。

单原型分类器，即最近均值分类器，是一种典型的模板匹配法，也是一种简单的基于相似度的分类器。其主要思想是用一个均值代表一个类别，分类时只需要比较新样本与各类均值的距离就可以确定属于哪一类。具体而言，在模型训练时，把训练样本按照类别分组，然后计算各类的算术平均值，用均值作为该类别的代表。在分类时，计算测试样本与各均值的欧氏距离，选取欧氏距离的均值所代表的类别作为测试样本的类别即可。

在图形分类器中，各类的代表性的多元图是均值的多元图，判别函数是图形相异度。各类的代

表性的多元图可以用几何均值、调和均值、中值的多元图等代替均值的多元图，判别函数也可以用方差加权距离、街区距离、马氏距离、夹角余弦和相关系数等代替图形相异度中的欧氏距离。

14.5.2　基于平行坐标的平行筛可视化分类方法

平行筛可视化分类器主要针对多类问题。其基本思想是首先用平行坐标表示原始训练数据，然后利用线性判别算法分别寻找各类最优投影方向（最优子空间）上的信息，最后采用决策树算法对各类别数据分别进行分割，最终形成分类筛图。根据分类筛图就可以对未知样本进行分类。

具体算法如下：

（1）数据预处理：包括归一化、缺失数据填补、离群点处理等。

（2）将预处理后的数据表示为平行坐标。

（3）寻找第 k 类的最优投影方向，$k=1,2,\cdots,K$。将第 k 类作为一类 $\boldsymbol{A}$，将其余 K-1 类作为另一类 $\boldsymbol{B}$，对 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 两类用线性判别算法求取最优的 v_k 个相互正交的投影方法，$v_k \geqslant 1$，这 v_k 个投影张成第 k 类的最优子空间。

类内离散度矩阵为

$$S_i = \sum_{x \in i}(x-m)(x-m)^T, i = A, B$$

总的离散度矩阵为

$$S_W = S_A + S_B$$

类间离散度矩阵

$$S_R = (m_A - m_B)(m_A - m_B)^T$$

分离度指标为

$$J = tr(S_W^{-1} S_B)$$

最优解为

$$w^* = S_w^{-1}(m_A - m_B)$$

（4）用平行坐标表示这 k 个最优子空间，其坐标轴数为 $\sum_{k=1}^{K} v_k$。

（5）用决策树算法在最优子空间 $\boldsymbol{Q}_k$ 中对第 k 类进行划分，确定分类规则。

（6）绘制决策树算法过滤后的平行筛图。

（7）根据平行筛图对未知样本进行分类。

14.5.3　基于平行坐标的贝叶斯可视化分类方法

首先将数据集随机分为测试集和训练集，然后将训练集用平行坐标表示。用参数化或非参数化方法对训练集数据各变量分别进行概率密度估计，据此进行非线性变换得到其点得分，最后对各变量的点得到进行加权融合，最终确定分类规则。

具体算法如下。

（1）数据预处理和降维。可采用各种通用降维方法。

（2）数据空间到图形空间映射。将 m 维数据空间的样本映射到二维平面的 m 个点，并用直线段或者曲线段连接属于同一个样本的 m 个点，就可以得到该数据样本点的坐标。

（3）分类优化。对平行坐标图进行处理和变换，使其更适合可视化分类需要。首先用参数化或者非参数化方法估计每维的类条件概率密度 $f_j(x|\omega_1), f_j(x|\omega_2)$，然后进行变换

$$x_{ij} \to \lg \mathrm{OR}(a_i) = \lg \frac{fj(xi_j|\omega_1)}{fj(x_{ij}|\omega_2)}$$

（4）绘制点得分平行坐标。用 n 条平行轴表示 n 个属性的点得分，每个样本用穿过 n 条平行轴的一条折线表示，折线与平行轴的交点的纵坐标对应属性的点得分值，用不同颜色，或者线条粗细，或者线条形状等区分不同的类别。

（5）分类规则确定。对点得分平行坐标中的各变量的点得分值进行加权融合，最终确定决策面参数和分类规则。分类规则为

$$\begin{cases} F(\omega_1|X) = \sum_i \lambda_i \lg \mathrm{OR}(a_i) > \mu, X \in \omega_1 \\ F(\omega_2|X) = \sum_i \lambda_i \lg \mathrm{OR}(a_i) < \mu, X \in \omega_2 \end{cases}$$

式中：X 为待分样本；λ_i 为权系数；μ 为判别点的值。权系数可以简单设置为等权，或者根据专家先验知识进行设置，以反映不同属性值的重要程度；OR（a_i）为第 i 个属性变量 a_i 对分类的贡献，即点得分

$$\mathrm{OR}(a_i) = \frac{P(a_i|\omega_1)}{P(a_i|\omega_2)} = \frac{\dfrac{P(\omega_1|a_i)}{P(\omega_2|a_i)}}{\dfrac{P(\omega_1)}{P(\omega_2)}}$$

类条件概率密度则可根据贝叶斯公式计算。

由于概率密度估计是在单变量上进行，所以计算复杂度较低，对样本数的需求也不太高，可以在人的监督下进行（如可以用直方图估计或者 K 近邻法），具有很好的可视化特性，从而有利于分类过程中专家知识参与以及对数据和分类结果的理解。

14.6　基于色度学空间的多元图表示

传统的多元图表示侧重于数据空间结构的表示，而对于数据的类别信息表示不足。为了引入类别信息，需要在不影响可视化的基础上，在传统多元图表示的维数上加入类别维以区分不同类别的分布情况。

目前常用的类别区分方法是将不同类别的数据表示为不同颜色或不同符号，虽然可以直观地观察不同类别数据在空间中的分布，但由于其定义每一个点只能对应为某种类别的颜色，因此对于重叠点数据却无法表示，而重叠点数据恰恰是分类器设计的重要参考点。

色度学认为：几种不同波长的光以一定比例的混合，可以得到一种全新的主观感受的颜色，该颜色的色度取决于参与混合的各颜色的比例。因此，在色度散点图中，可以根据类别数目选择适当的基色进行类别的标识。在非重合点，色度直接用基色表示，而在重合处，若为同类别重合，

则基色与自身混合，仍为原始基色，不影响信息表示；若为不同类重合点，则根据该点上类别的概率分布对不同类别的基色进行混合，得到的混合色用于当前点的着色。在着色时，因为关心的是类别的概率而不是绝对数目，所以需要对空间中相同坐标点的不同类别数据做归一化处理。

14.7　基于 MATLAB 的数据可视化技术

例 3.26　监测某湖泊的水质，共设 7 个监测点，每个监测点监测指标为 5 项，监测结果如表 14.1 所示。试用各种可视化方法表示之。

表 14.1　水质监测数据表　　单位：mg/l

点　位	DO	COD	BOD	T－N	T－P
1	4.3	4.74	4.23	3.66	0.105
2	5.9	4.61	2.59	2.92	0.081
3	7.0	3.94	2.92	1.71	0.072
4	6.9	3.92	3.11	1.32	0.075
5	7.4	4.02	3.10	1.26	0.076
6	6.9	3.75	3.15	1.05	0.096
7	6.7	4.44	3.14	1.02	0.072

解：

```
>> load mydata;                                          %输入数据
>>parallelplot(x,{'DO','COD','BOD','T-N','T-P'});        %画图，得图14.15
>> triangleplot(x);                                      %三角多项式图（图14.16）
```

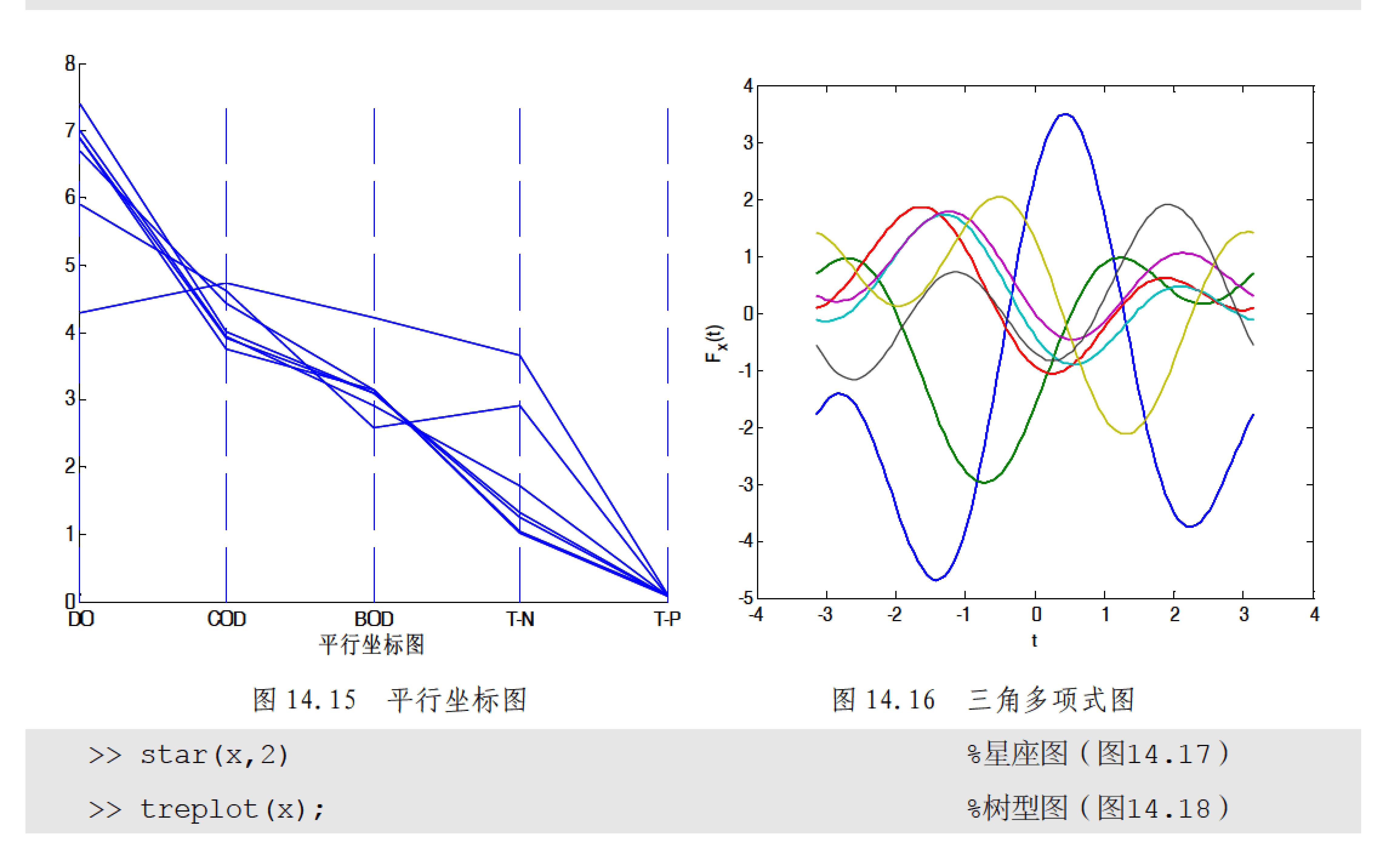


图 14.15　平行坐标图　　　　图 14.16　三角多项式图

```
>> star(x,2)                                             %星座图（图14.17）
>> treplot(x);                                           %树型图（图14.18）
```

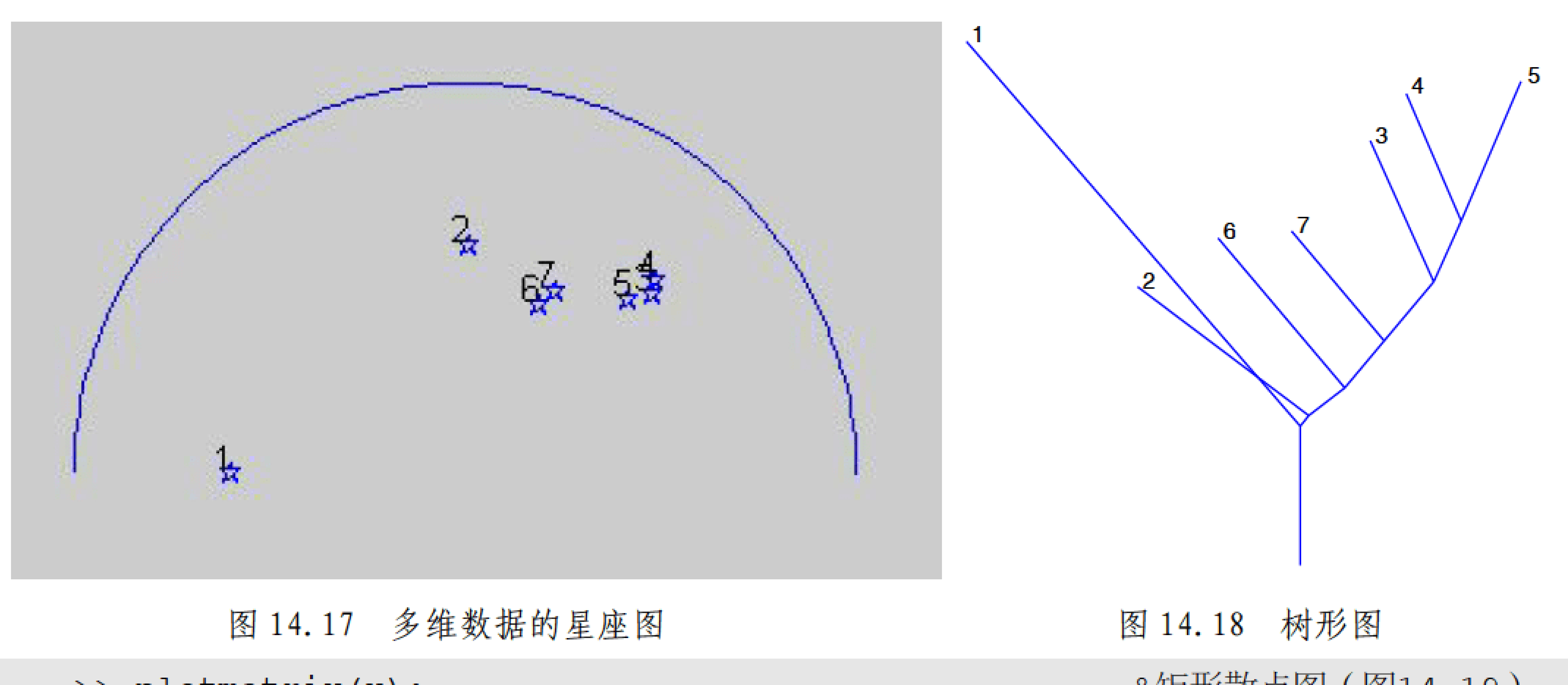


图 14.17　多维数据的星座图　　　　图 14.18　树形图

```
>> plotmatrix(x);                        %矩形散点图（图14.19）
>>y=zigzag(x,type,num);                  %折线图（图14.20）
```

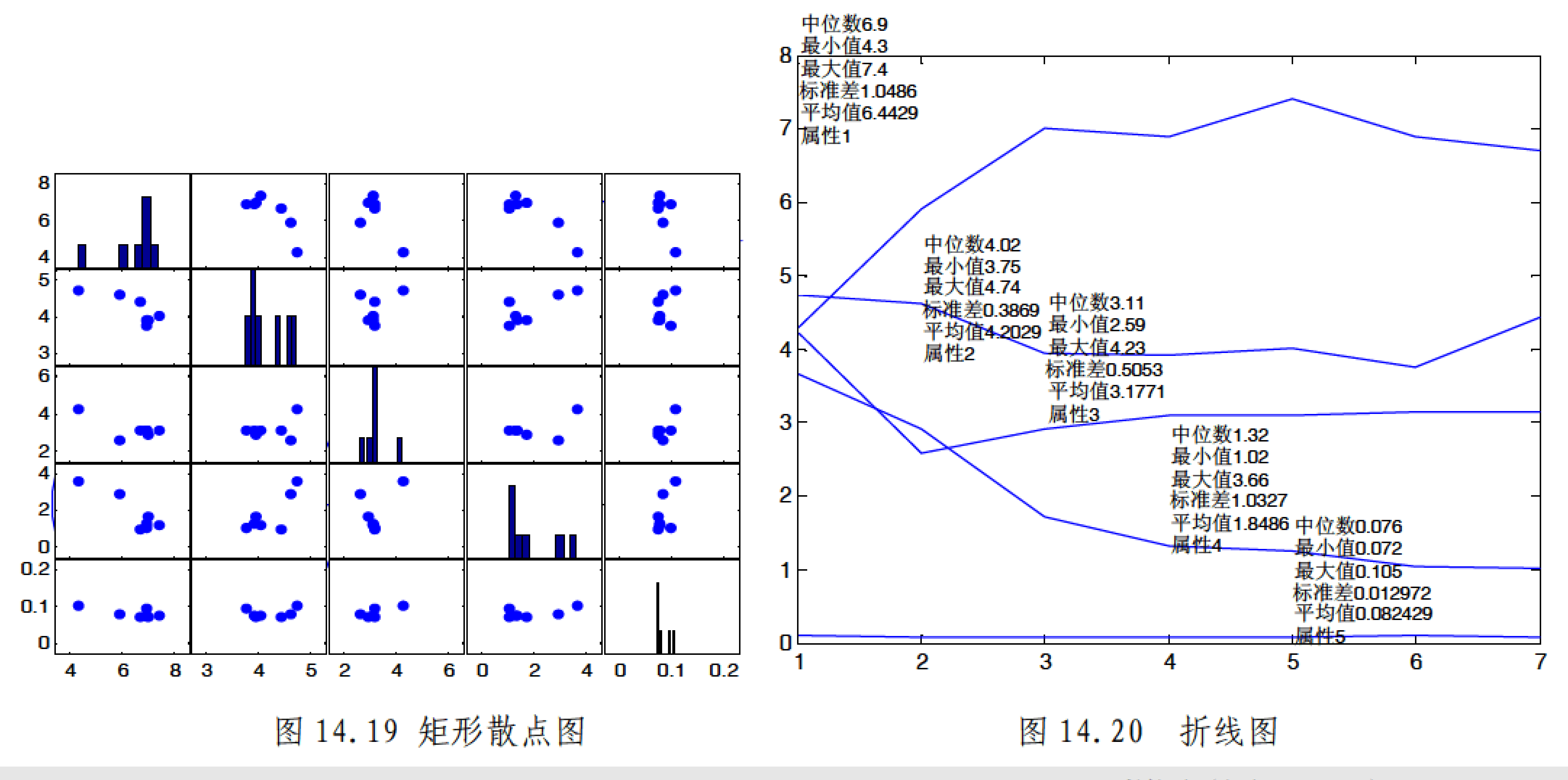


图 14.19 矩形散点图　　　　图 14.20　折线图

```
>>y=mybar(x,3);                          %饼图（图14.21）
```

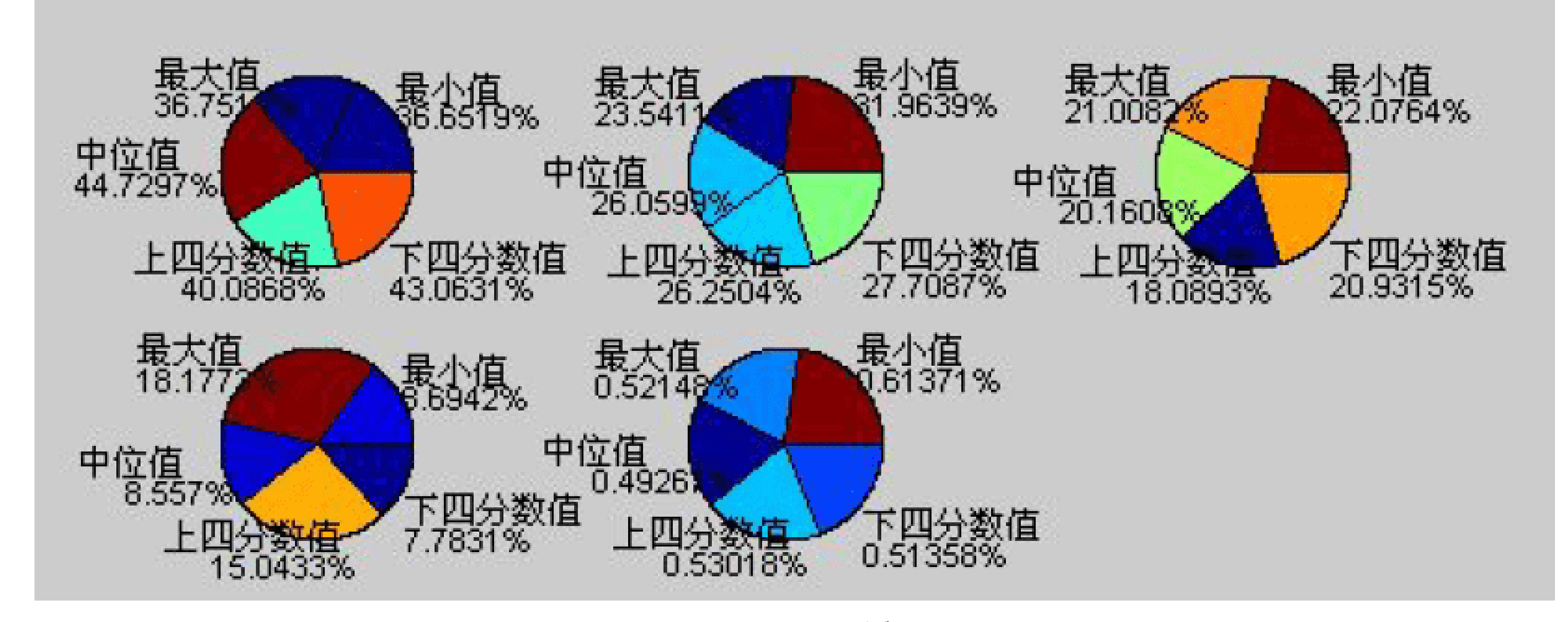


图 14.21　饼图

```
>>y=cirqueplot(x);                       %圆环图（图14.22）
>>y=star_coordinate(x,c1,alpha,theta1,x1)   %星型坐标图（图14.23）
```

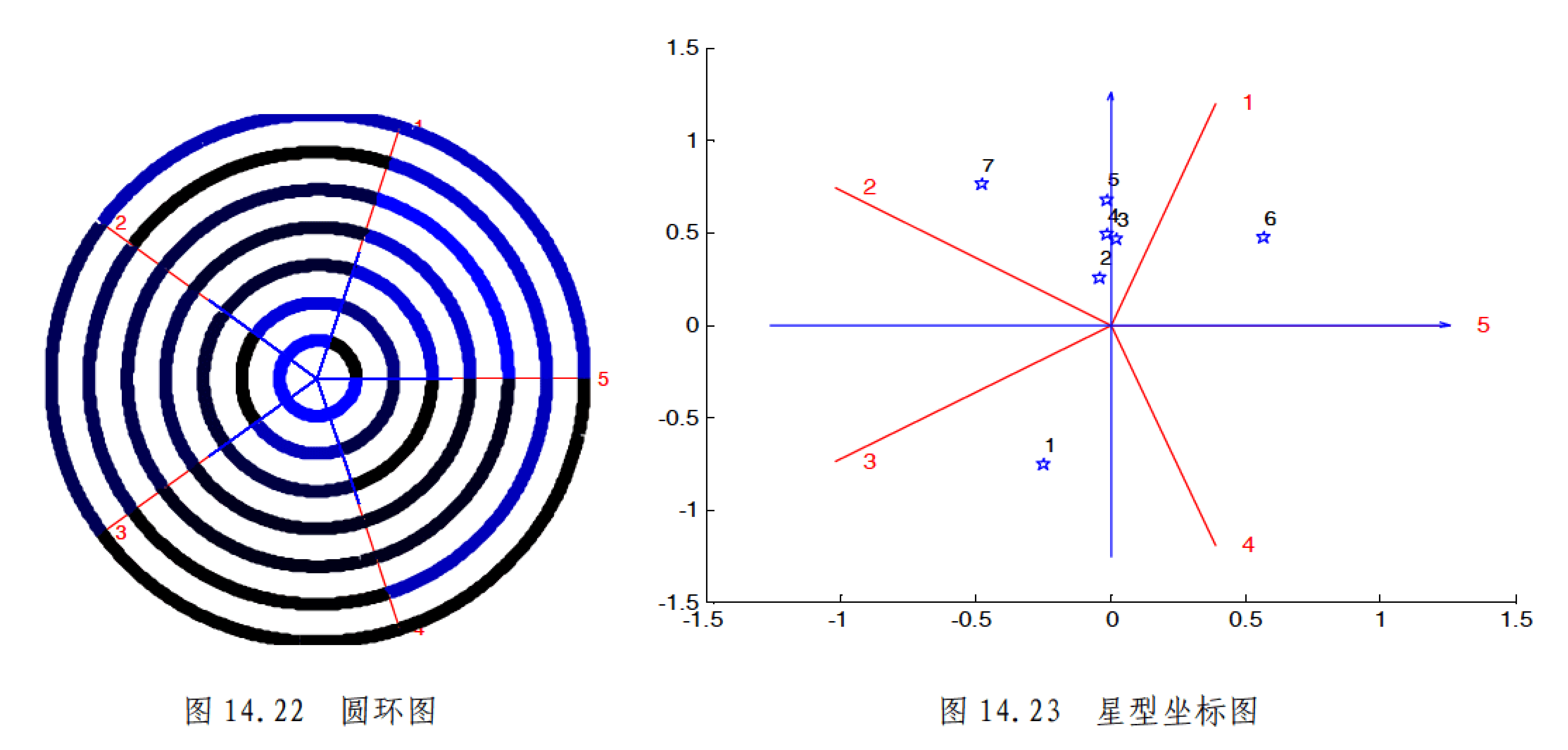


图 14.22　圆环图　　　　　　　　　　图 14.23　星型坐标图

例 3.27　利用极坐标映射的方法表示 Iris 数据。

解：

根据极坐标映射的原理，可编程计算，其中参数默认值为中心坐标（1,1），$f(x)=x;g(x)=x;k=1$。用户也可以自行设定。

```
>> a=dlmread('D:\数据.txt');                %读入数据
>> x=a(:,1:4);                              %数据中有一列为零，删除
>> polarMap(x(:,1),1)                       %图 14.24，其余变量类似，不再列出
```

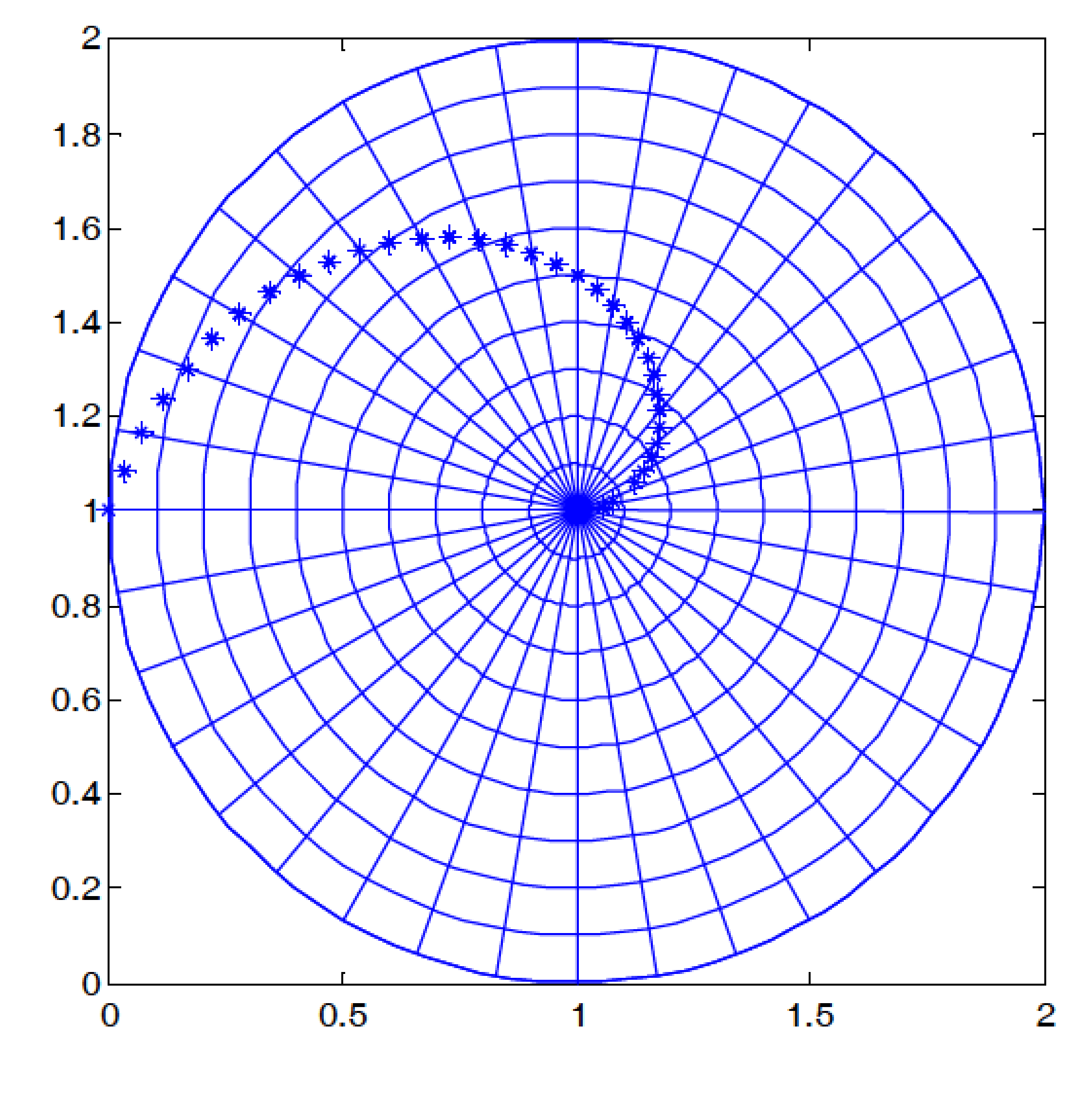


图 14.24　花瓣长度的极坐标映射图

例 3.28　对于高维数据，除了利用例 14.23 中的各种方法进行可视化显示，还可以采用诸如主成分分析、投影寻踪方法、非线性映射等方法进行降维以使其在维空间中显示。下面利用这三种方法对表 14.2 中的数据进行降维处理。

表 14.2　15 个标准中国茶叶样品的化学成分

样　品	浓度（%w/w）					
	纤 维 素	半纤维素	木 质 素	茶 多 酚	咖 啡 因	氨 基 酸
1	9.50	4.90	3.53	29.03	4.44	3.82
2	10.06	5.11	3.57	27.84	4.29	3.70
3	10.79	5.46	4.62	26.53	3.91	3.46
4	10.31	4.92	5.02	25.16	3.72	3.29
5	11.50	6.08	5.48	23.28	3.50	3.10
6	12.10	5.64	5.61	22.23	3.38	3.02
7	13.30	5.68	6.32	21.10	3.14	2.87
8	9.07	5.33	4.42	27.23	4.20	3.18
9	10.75	5.80	5.29	25.99	4.00	3.00
10	10.78	5.72	5.79	24.77	3.86	2.91
11	12.00	6.68	7.20	24.05	3.49	2.81
12	12.17	5.86	7.71	23.02	3.42	2.60
13	10.32	10.66	5.07	21.55	4.23	4.43
14	10.99	10.11	5.60	20.64	4.14	4.35
15	12.32	10.12	6.53	20.06	4.02	4.12

解：

以下各程序中的“num”均表示为显示样本序号样本数的阈值。

（1）非线映射方法。

设有高维数据点 $X_i(x_{i1},x_{i2},\cdots,x_{im})$，其二维显示的对应点是 $Y_i(y_{i1},y_{i2})$，则 Y_i 是 X_i 的某种函数，如果 y 是各 x 的某一线性组合，则二维图像是高维图像的投影。如果 y 和 x 间的关系是非线性函数，则二维图像是高维图像的非线性映射（Non-linear Mapping，NLM）。

根据非线性映射方法，映射时的误差函数为

$$E = f(d_{ij}^* - d_{ij}) = \frac{1}{\sum_{i<j}^{n} d_{ij}^*} \sum_{i<j}^{n} \frac{[d_{ij}^* - d_{ij}]^2}{d_{ij}^*}$$

其中：d^*_{ij}、d_{ij} 分别为高维数据和二维数据的欧氏距离。据此可利用遗传算法对该函数进行最小化处理，找到合适的二维数据结构，完成高维数据到二维数据的非线性映射。

根据非线性映射的方法原理，可编程计算得到如图 14.25 所示的结果。

```
>> load data;
>> y=myNLM(data,40);
```

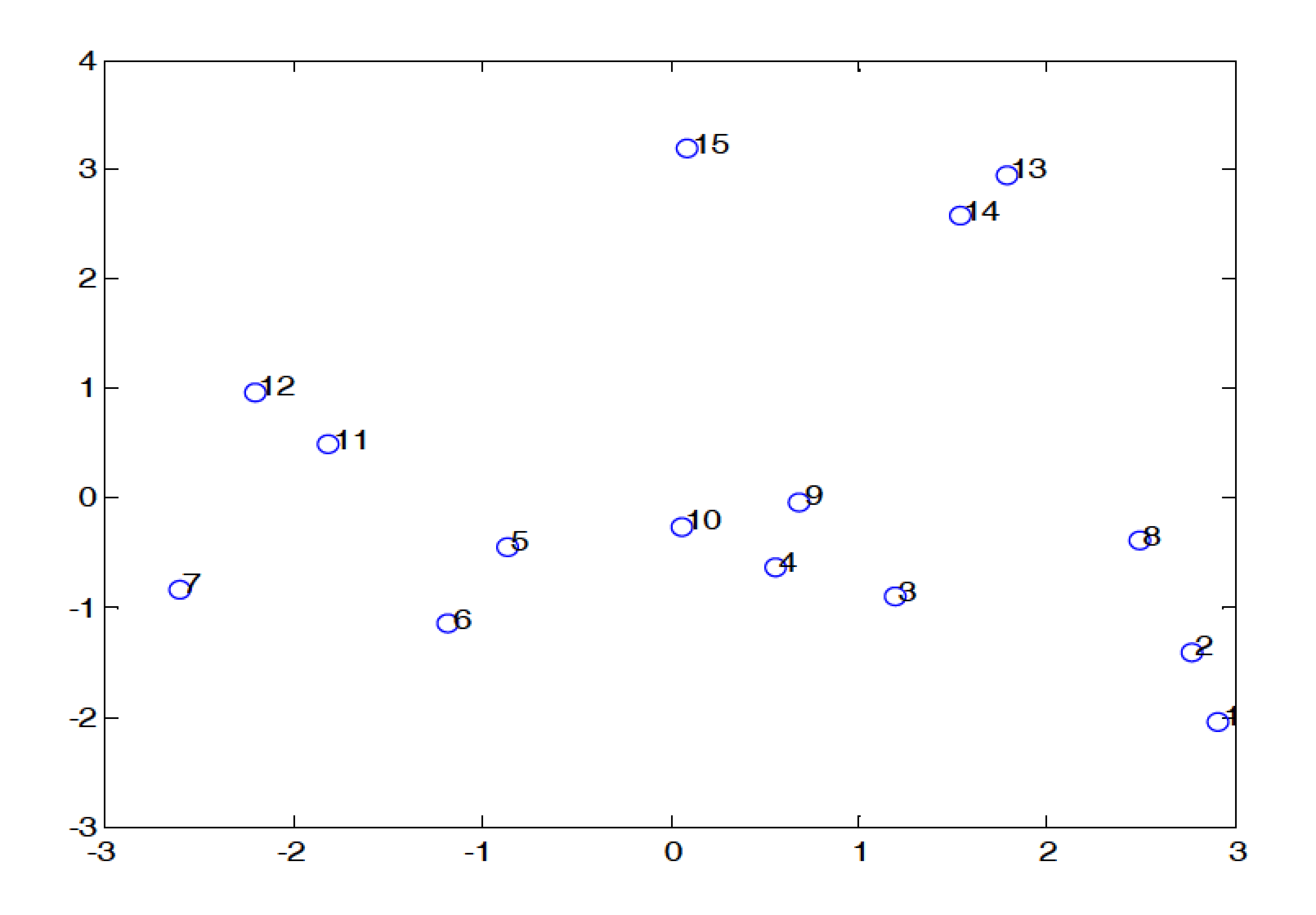


图 14.25　原始数据非线性映射结果的图像

（2）主成分分析方法。

根据主成分分析方法原理，可编程计算得到如图 14.26 所示的结果。

```
>> [y,num]=myprincomp(data,40);
```

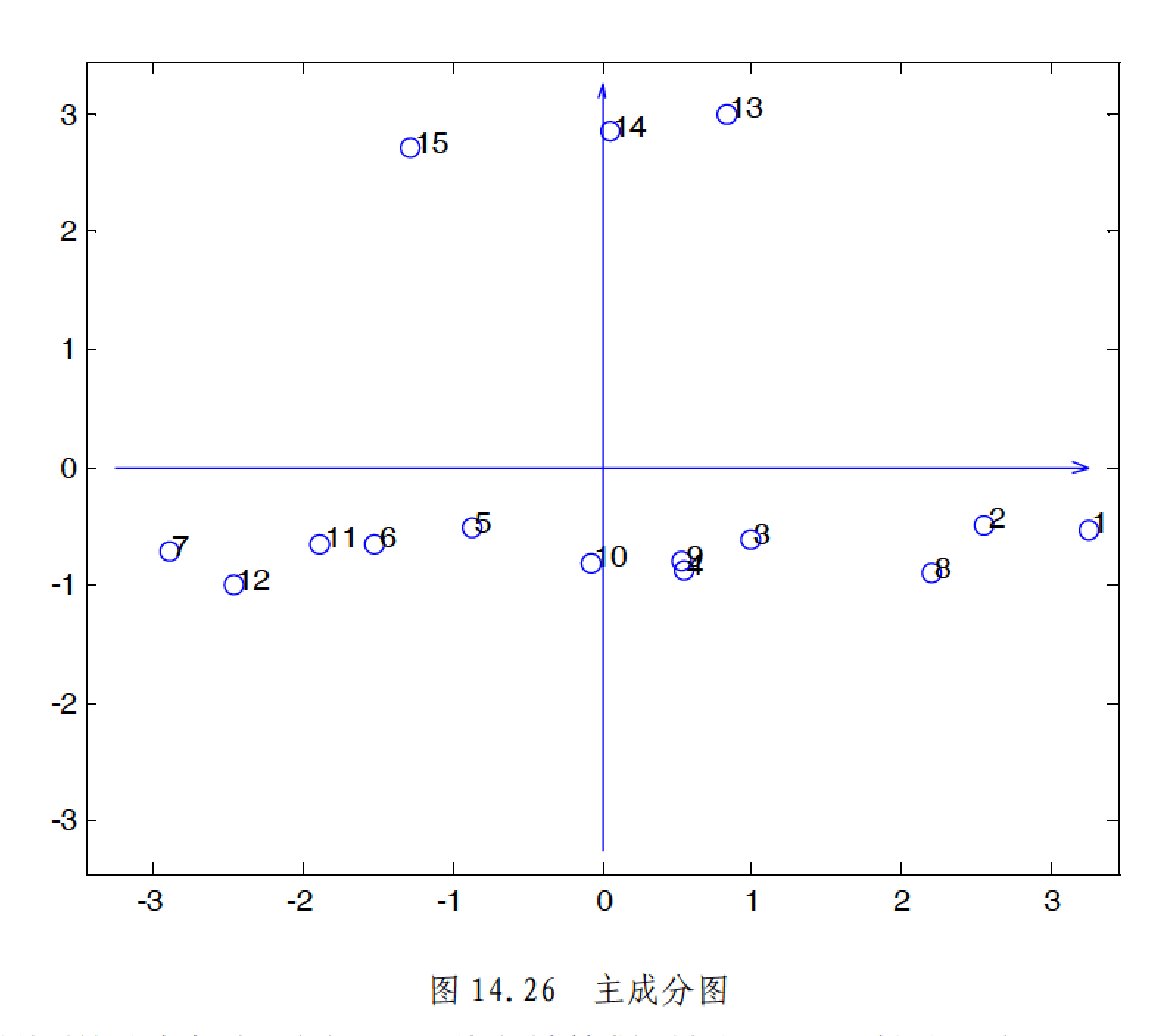


图 14.26　主成分图

（3）根据投影寻踪方法的原理，可编程计算得到如图 14.27 所示的结果。

```
>> y=myPP(x,40);
```

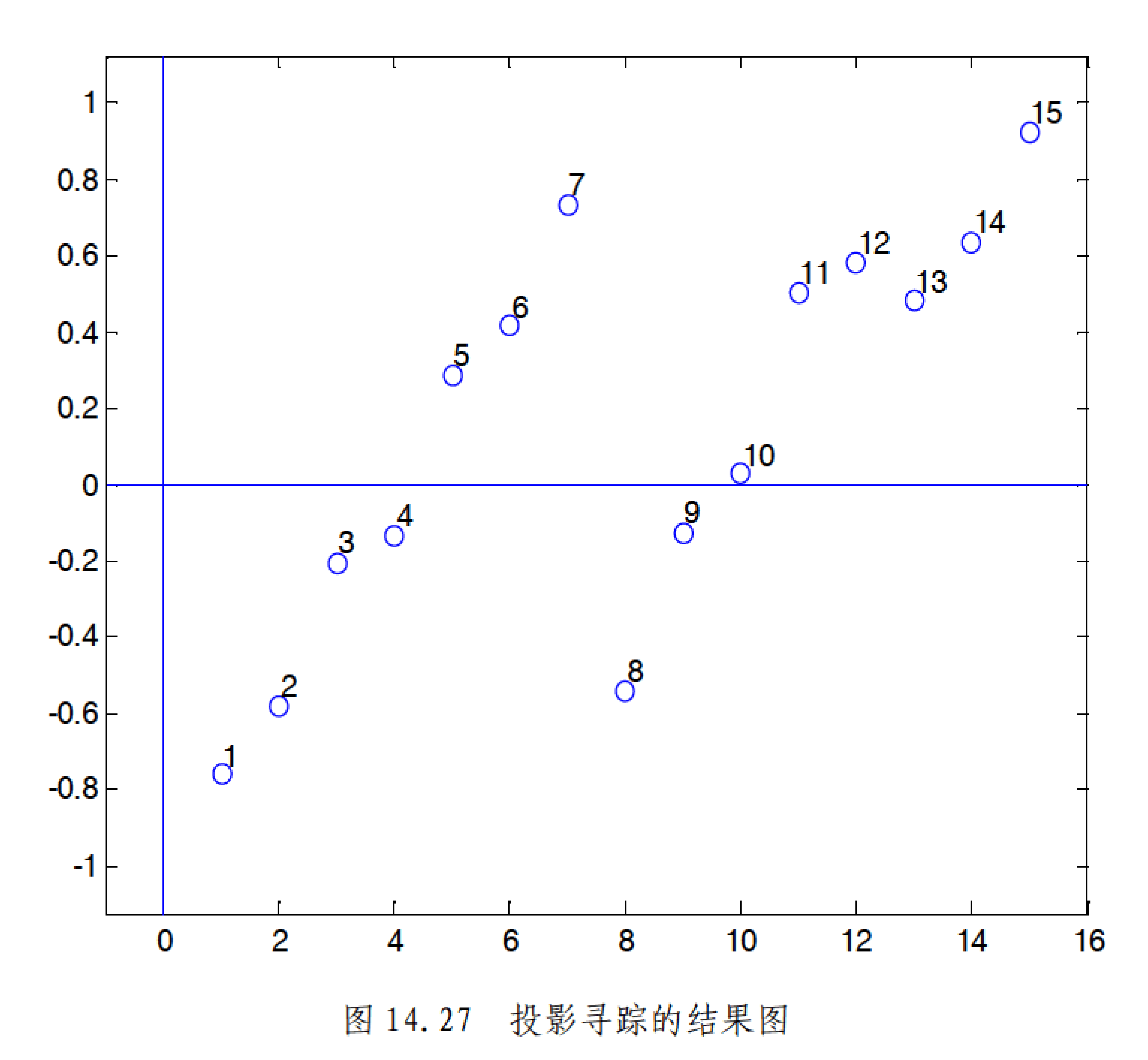


图 14.27　投影寻踪的结果图

要注意的是，程序中数据归一化对于不同的应用有不同的计算方法。如在应用投影寻踪进行评价时，对于越大和越小的指标归一化方法就有所不同：

对于越大越优的指标，$x_{ij}^* = \dfrac{x_{ij} - x_{\min}(j)}{x_{\max}(j) - x_{\min}(x)}$

对于越小越优的指标，$x_{ij}^* = \dfrac{x_{\max}(j) - x_{ij}}{x_{\max}(j) - x_{\min}(x)}$

例 3.29　数据挖掘中各种规则的可视化表示也是经常会遇到的问题。现用平行坐标法对以下规则进行可视化表示。

```
rule={'I1,I2→I3' 0.5;'I2,I3→I1'  0.5;'I3,I1→I2'  0.5;'I1,I2→I5'  0.5;
     'I2,I5→I1'  1.0;'I1,I5→I2'  1.0;'I1,I3→I2,I4,I5' 0.8;'I5→I1,I2'  1.0};
```

解：

规则的可视化表示最为常见的是平行坐标法。根据其原理，可编程绘制如图 14.28 和图 14.29 所示的结果。

```
>> parallel_ass(rule,1)     %图 14.28，其中箭头后面的项为规则的后件
>> parallel_ass(rule,2)     %图 14.29，其中圆表示规则的前件，矩形表示为规则的后件，
                             矩形填充颜色表示可信度
```

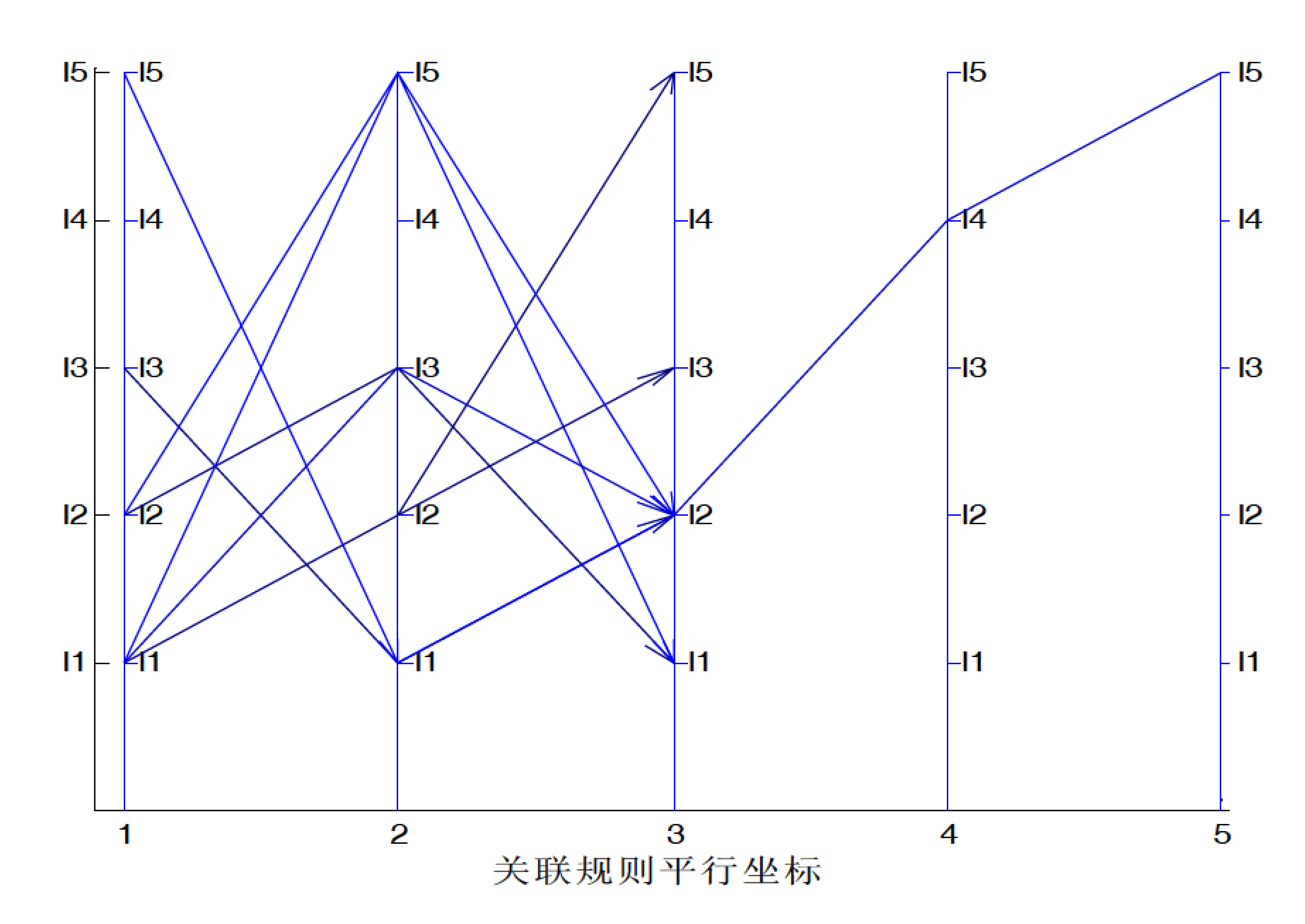


图 14.28　规则的可视化表示图（一）

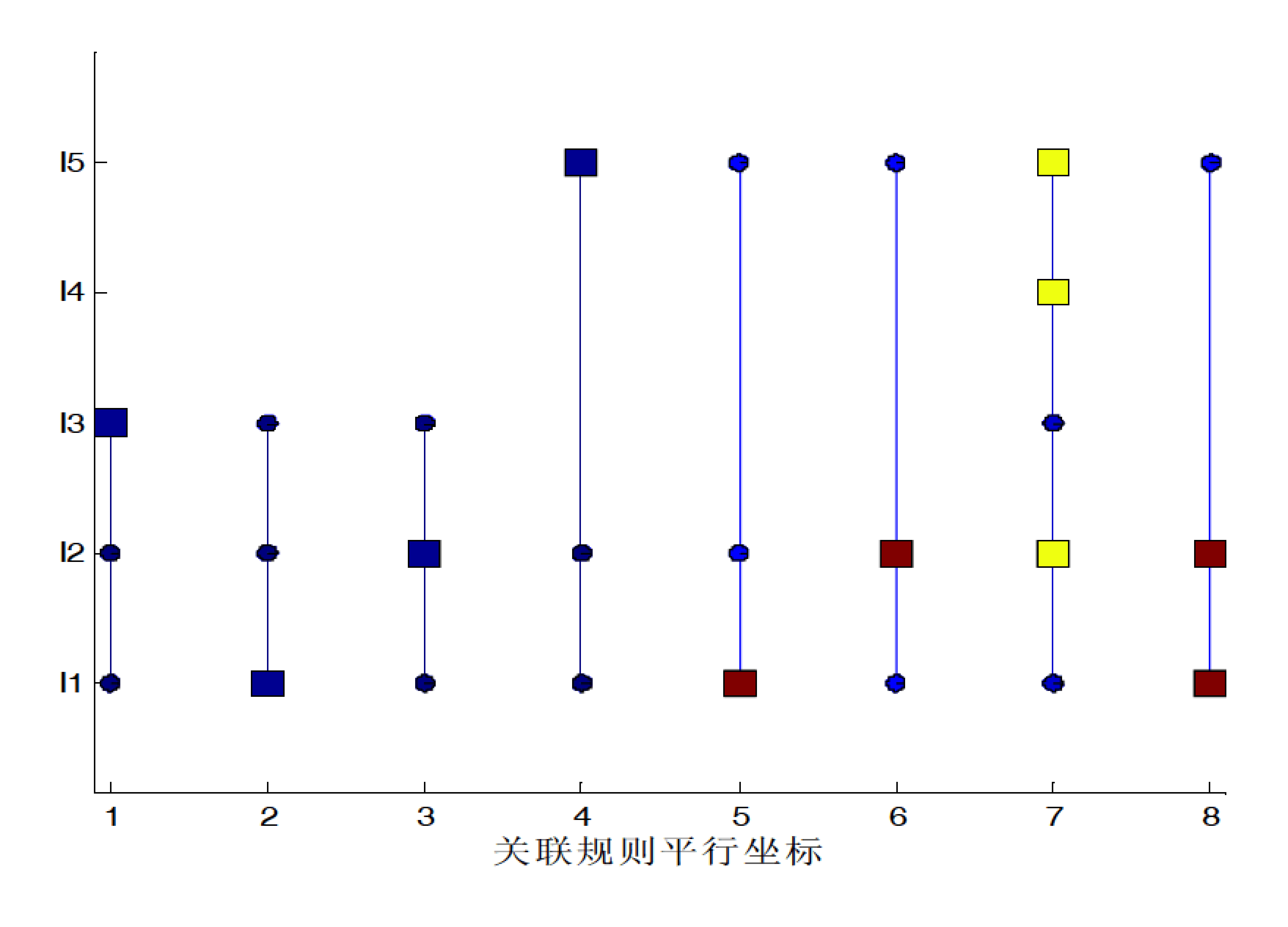


图 14.29　规则的可视化表示图（二）

例 3.30　规则除了例 3.29 中的单维形式，还有如下所示的多维规则，对此同样可以用平行坐标法表示。

```
rule={'景色=多雨∧湿度=正常∧风力=无→活动=可以',0.6
      '景色=多云∧风力=大∧活动=不可以→温度=温',0.5
      '温度=热∧湿度=高∧风力=小∧活动=可以→景色=多云',0.8};
```

解：

```
>> parallel_ass1(rule)    %得图14.30
```

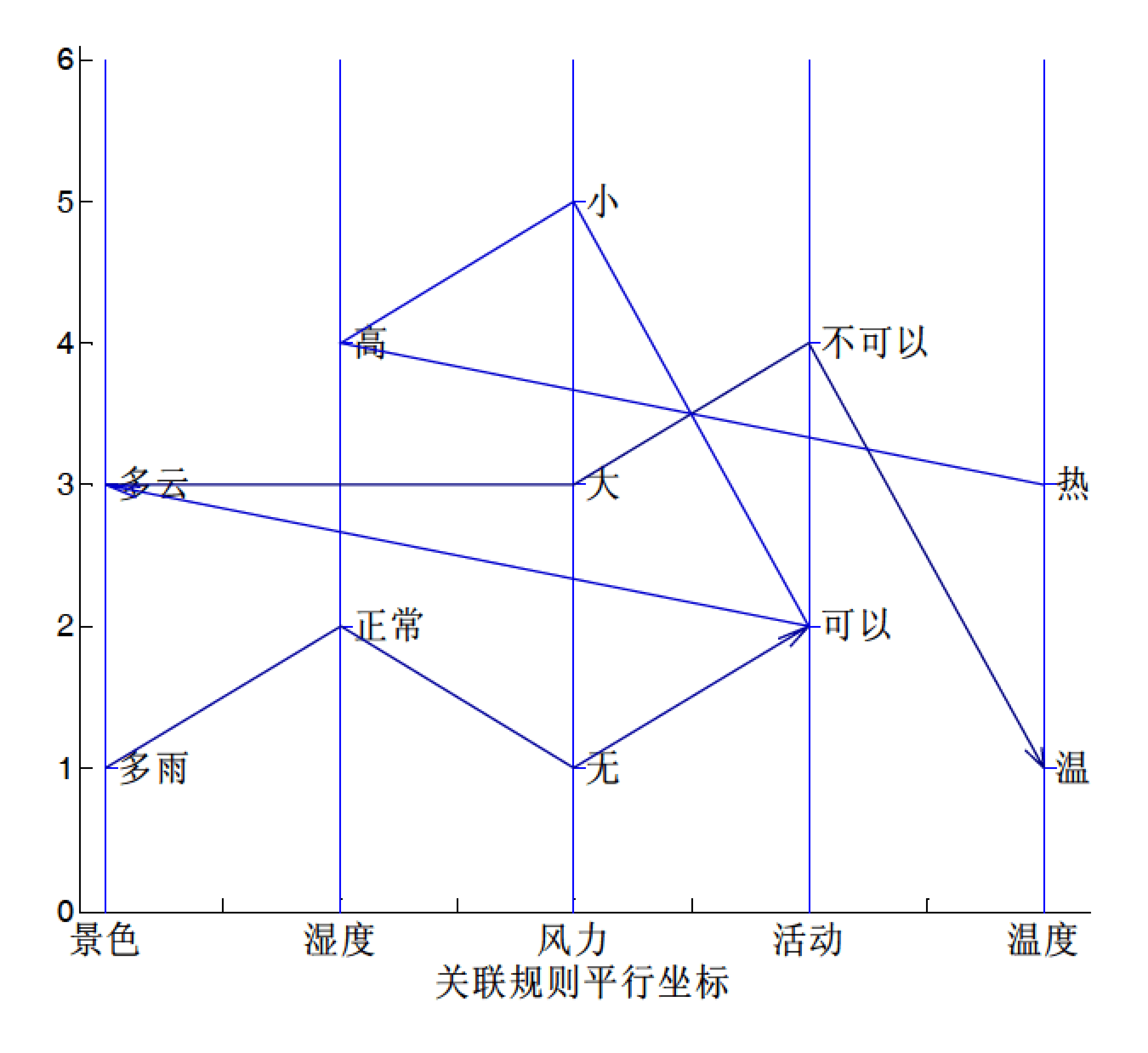


图 14.30　多维规则的可视化表示图

例 3.31　规则的可视化表示还可以用柱形图。用此法表示以下规则：

```
rule={'I1,I2→I3' 0.5;'I2,I3→I1'  0.5;'I5→I1,I2'  1.0};
```

解：

```
>> parallel_ass(rule,3);    %得图14.31
```

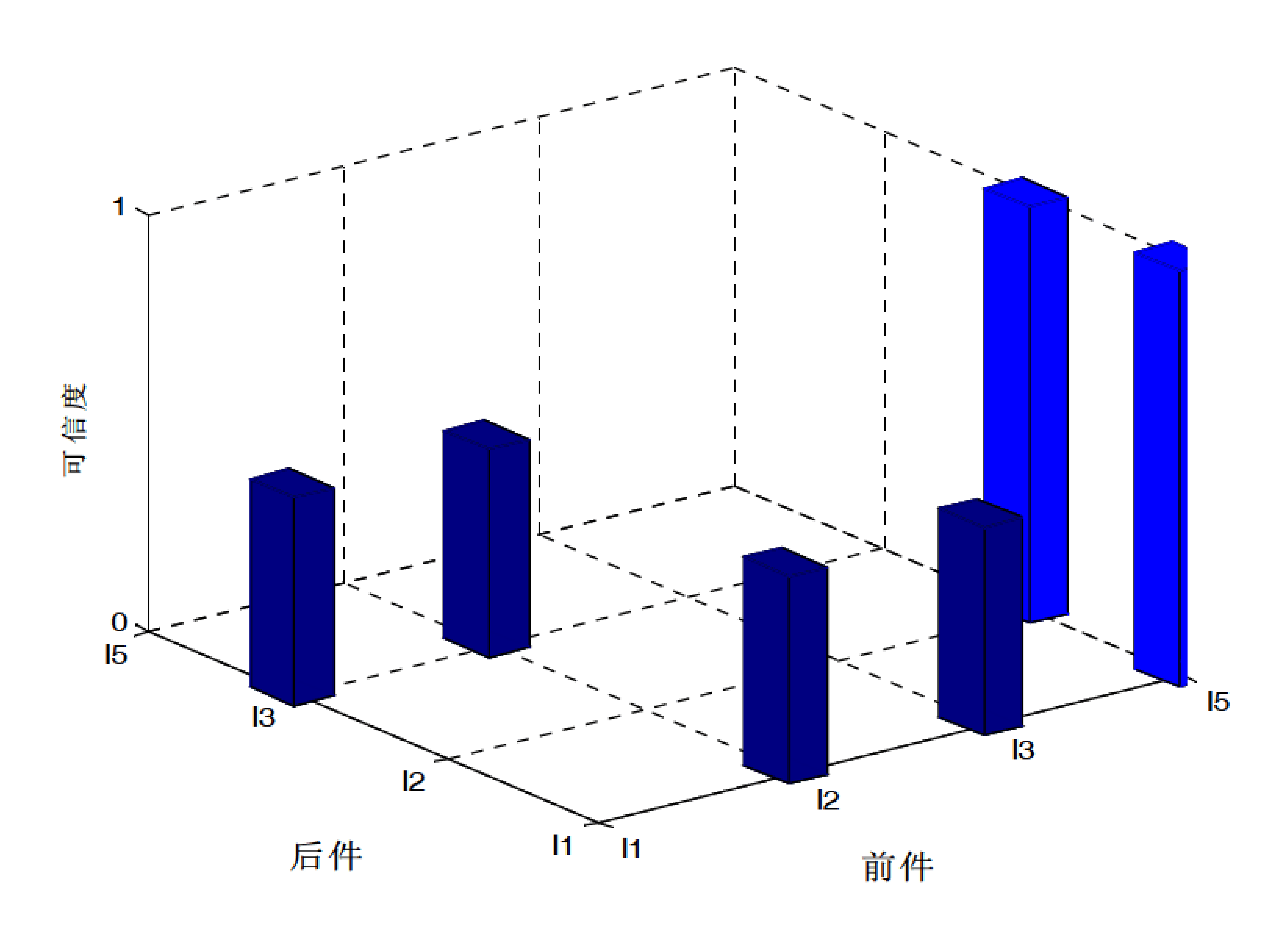


图 14.31　规则的可视化表示柱形图

例 3.32　利用雷达图等图形特征也可以对样品进行分类。试利用雷达图的基于重心的图形特征对 Iris 数据进行分类分析。

解：

因为雷达图的形状与特征排序有关，而固定特征排序下的样本对应唯一的雷达图。一旦特征排序确定，从雷达图中提取出重心图形特征就仅与这个固定的特征排序有关。所以在利用图形特征分类时，首先要确定特征的排序。

可以用遗传算法来求出较佳的特征排序，其参数设置如下。

种群数：30；

迭代次数：100；

变异概率：0.05；

交叉概率：0.85；

编码长度：特征数，本例中为4，编码方式为1~4的正整数，如[2 4 1 3]，编码顺序即为数字对应的特征的排序。

适应度函数：图形特征分类器对样本分类的正确率。其中随机采用2/3的样本作为训练集，其余为测试集，计算测试集分类结果的正确率。

根据以上参数，就可以编程进行计算。编程时要注意的是基因经过变异、交叉操作后，会出现不合理的编码（即出现重码以及缺码），因此在编变异及交叉函数时需要进行处理，以防止这类情况的产生。

计算结果如下：

```
>> a=dlmread('D:\数据.txt');
>> x=a(:,1:4);
>>m=30;t=100;pc=0.85;pm=0.05;class=[ones(50,1);2*ones(50,1);3*ones(50,1)];
>>y=graph_ga(x,m,t,pc,pm,class);
```

其中一次的计算结果：

```
y =value: [4  3  1  2]    %迭代6次的结果
   fit: 1
```

此排列顺序，也可以根据spearman系数进行对比。此例中的spearman系数如下。

```
>> y=spearman(x)                    %负数表示负相关
y =  1.0000     -0.1608     0.8821     0.8352
      1.0000     1.0000     -0.3027     -0.2773
      1.0000     1.0000     1.0000     0.9380
      1.0000     1.0000     1.0000     1.0000
```

例 3.33　基于图形相异度的图形分类器可以用于未知样本的分类。试利用单原型图形分类器对 Iris 数据中的数据进行分类分析，以检验方法的适用性。

解：

选择Iris数据集中适量的三类样本作为训练样本，适量的样本作为测试样本。然后根据模板

匹配法原理进行测试样本的分类。

据此，可编程计算如下。

```
>>a=dlmread('D:\数据.txt');
>>x=a(:,1:4);
>>train=[x(1:10,:);x(51:60,:);x(101:110,:)];
>>class1=[ones(10,1);2*ones(10,1);3*ones(10,1)];
>>sample=[x(12:15,:);x(63:65,:);x(121:123,:)];
>> y=gram_classify(train,sample,class1);    %分类完全正确
y =1     1     1     2     2     2     3     3     3
```

例 3.34　径向坐标可视化，也称为弹簧力模型是 Hoffman 等人于 1997 年提出的一种多维数据可视化表示方法。其基本思想是在二维平面上的一个圆内部将所有样本以点的形式表示出来，而样本的特征值均等分布在圆周上。该方法可以用实际中的弹簧力平衡物理模型来理解：映射点连接多根弹簧的一端，各弹簧的另一端与均等分布在圆周的特征点连接，某根弹簧的刚度为样本在该特征的定量值，映射点稳定于各弹簧合力为零处。

试用径向坐标可视化方法表示Iris数据。

解：

径向坐标可视化方法的原理如图14.32所示。据此以及力平衡原理，便可以编程计算映射点的坐标，并画出其图像。

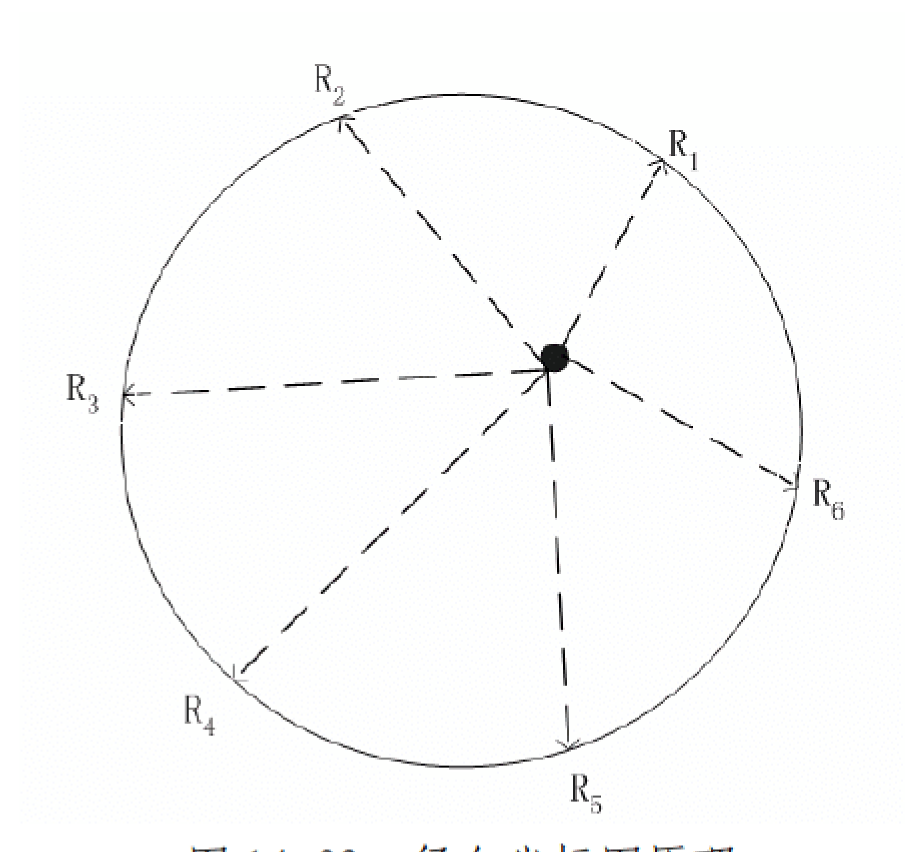


图 14.32　径向坐标图原理

```
>> a=dlmread('D:\数据.txt');
>> x=a(:,1:4);
>> class=[ones(50,1);2*ones(50,1);3*ones(50,1)];
>> radviz(x,class)
```

作图时可以将圆画出，也可以不画出，如图14.33所示。

另外，可以通过函数中的type参数对传统的径向坐标图进行改进（此时为's'，传统的为'n'），即在1/4圆周上将维数等分，这样可使点更分散，如图14.34所示。

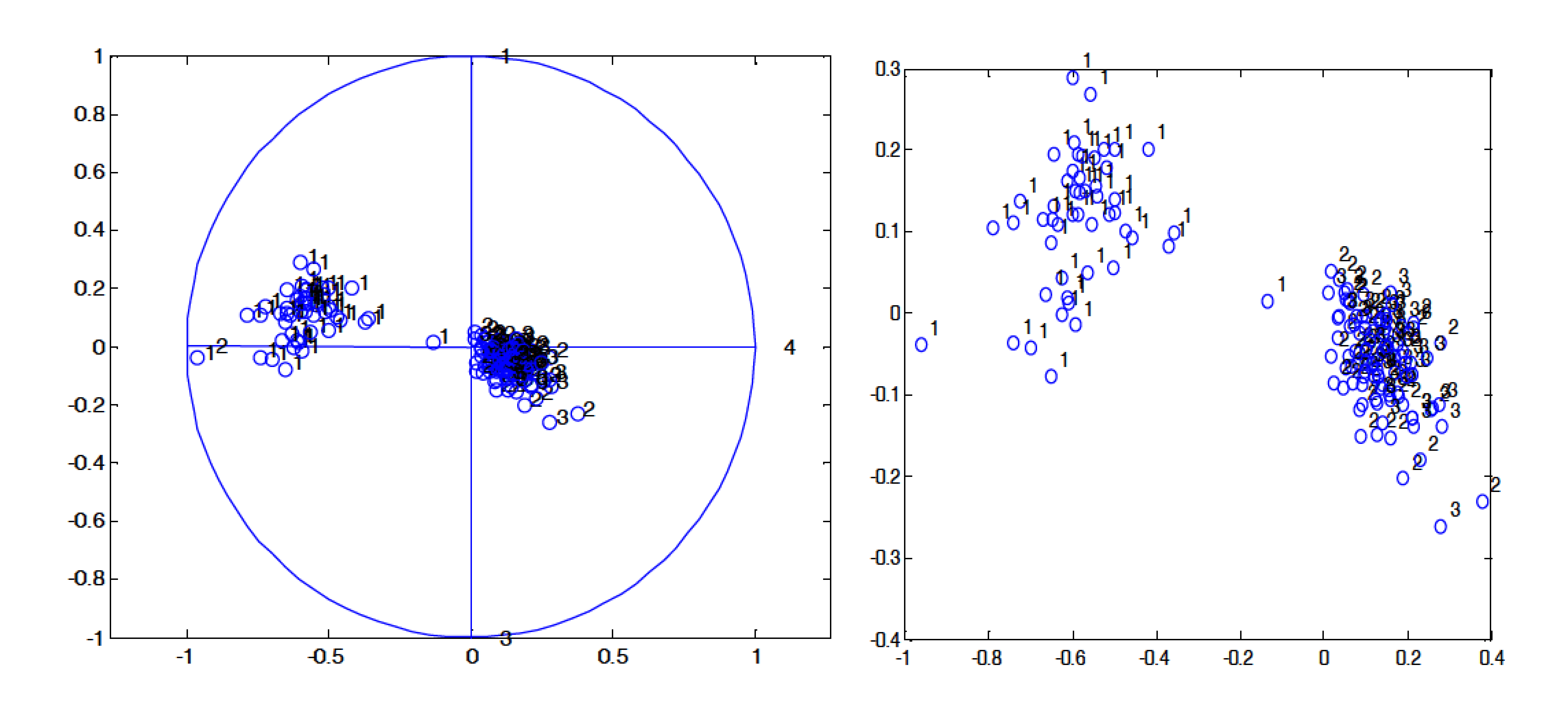


图 14.33　Iris 数据的径向坐标图

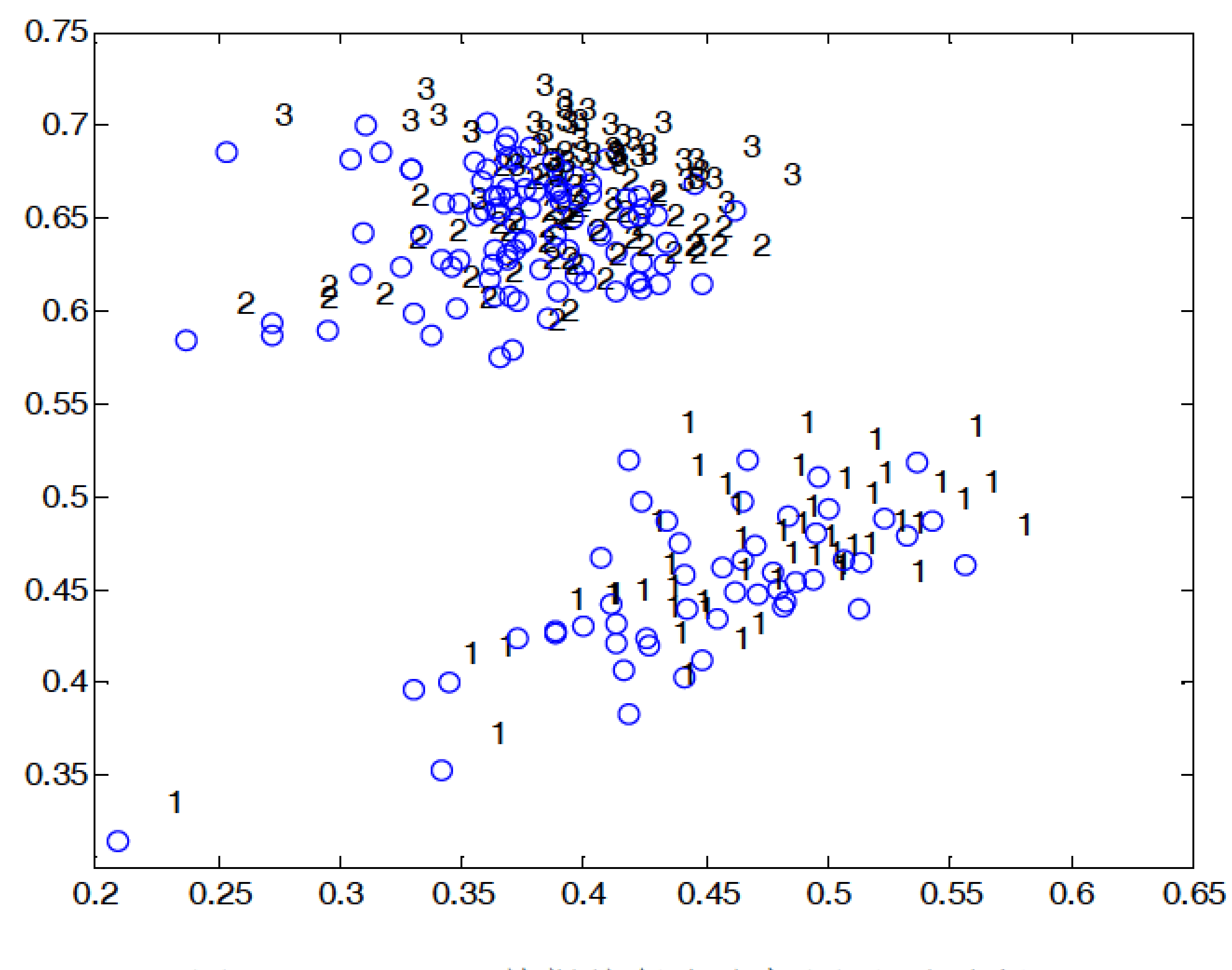


图 14.34　Iris 数据的径向坐标图（1/4 圆）

例 3.35　利用色度学原理也可以表示高维数据集。颜色是视觉系统中可见光的感知结果，可见光是波长在 400~700mn 之间的电磁波，人眼所能感受到的只是波长在可见光范围内的光波信号，当各种不同波长的光波信号一同进入人的眼睛的某一点时，人的视觉器官会将它们混合起来，作为一种颜色接受下来。因此，如果将多维数据集中的每维数据当作一段波长的光，则经过一定的处理（如同颜色的混合），类似于三基色坐标系表示一种颜色一样，就可以在三维坐标图中将多维数据表示出来。

试用此方法表示Iris数据集。

解：

对于多维空间中的每一个数据点$\boldsymbol{X}$，可以将其看作是一个彩色刺激函数为$x_i(\lambda)$，均匀地分布在波长为400~700mn的可见光波长范围内，即$x_i(400)=x_{i1}$，$x_i(700)=x_{ip}$，从而得到x_i对应波长的函数关系$x_i(\lambda)$，再利用光谱响应函数对彩色刺激函数的转换就可以推出数据x_i的三维R、G、B坐标：

$$R = k \times \sum_i X_i(\lambda) r(\gamma) \Delta\lambda_i$$

$$G = k \times \sum_i X_i(\lambda) g(\gamma) \Delta\lambda_i$$

$$B = k \times \sum_i X_i(\lambda) b(\gamma) \Delta\lambda_i$$

式中：k为比例系数；r、g、b分别为光分布色系数（由表可查），如图14.35所示。

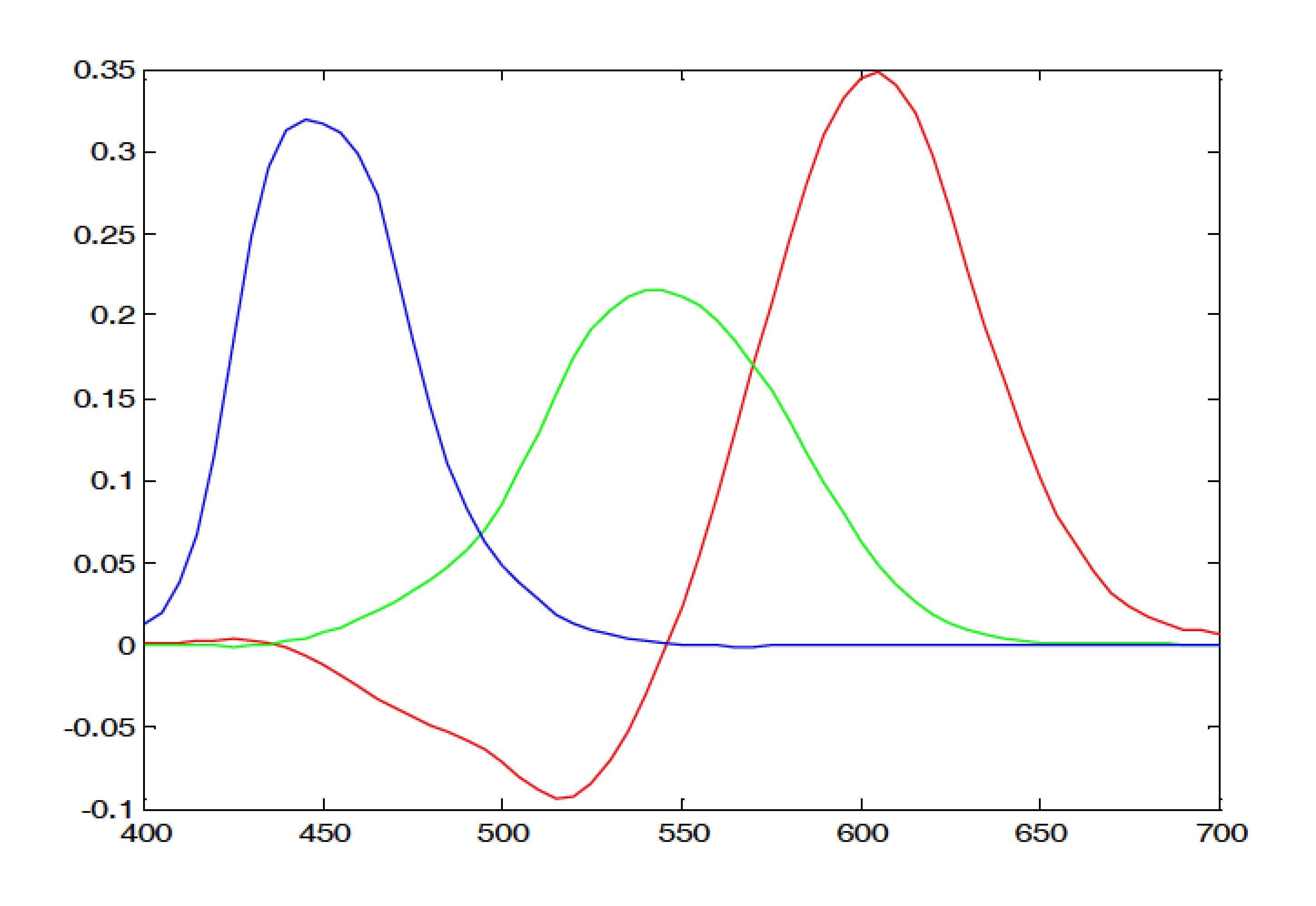


图 14.35　光分布色系数

据此，可编程计算发如下。

```
>> a=dlmread('D:\数据.txt');
>> x=a(:,1:4);
>> chrogram(x);      %得图14.36
```

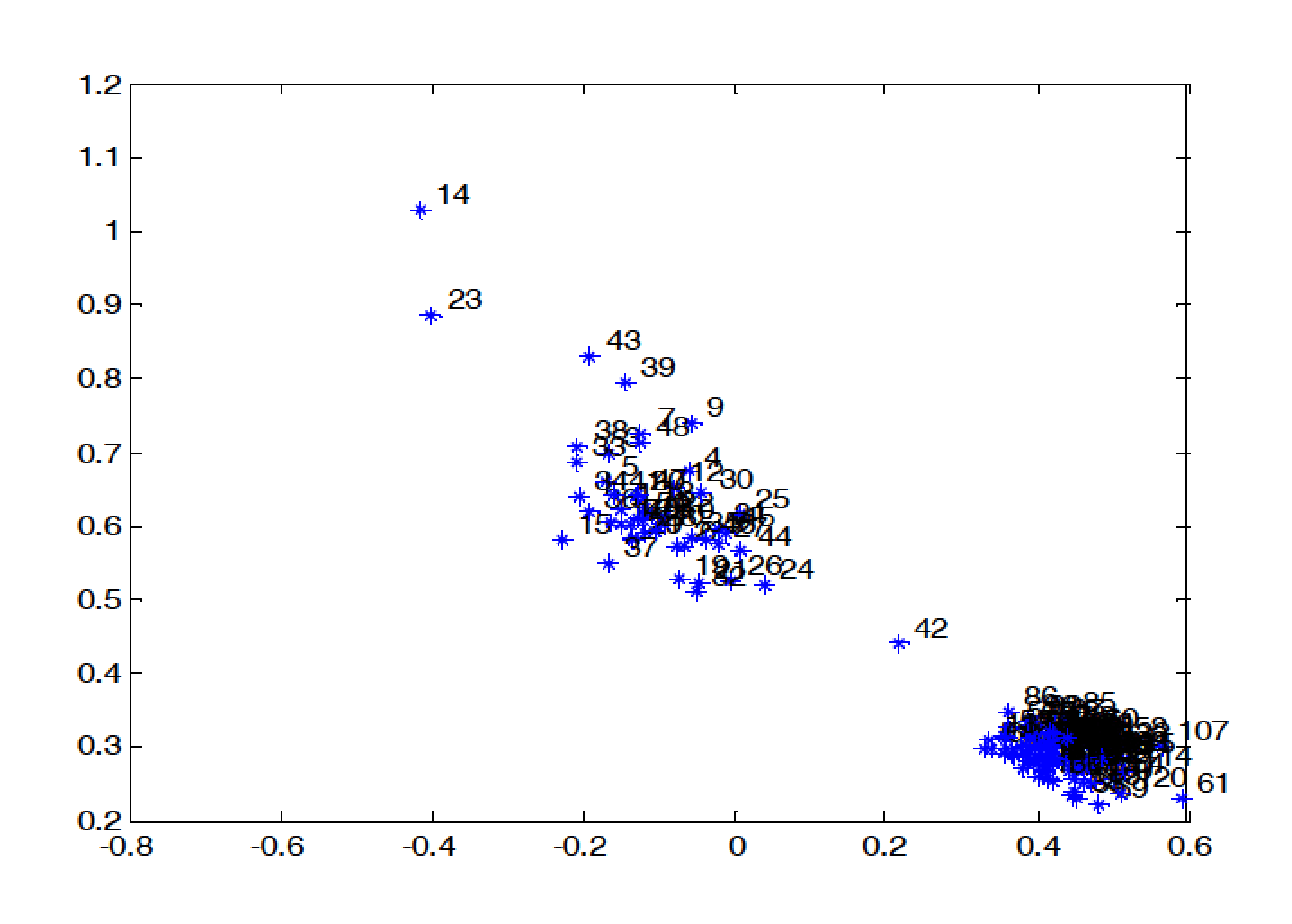


图 14.36　Iris 数据的图像表示

从图中可看出，数据可以分成两簇，其中一簇为第一类，另外一簇为二、三类样品，与用径向坐标表示的结果类似。

例 3.36　脸谱图是用脸谱来表达多变量的样品。一个人的脸谱可以具有非常生动的表情及形象，脸的胖瘦、喜怒哀乐给人留下深刻的印象。用脸谱来表达多变量首先是由美国统计学家 H.Chernoff 于 1970 年提出的，他将样品的 p 个变量用人脸的某一个部位的形状或大小来表示，一个样品用一张脸谱来表达。他首先将脸谱图用于聚类分析之中，引起了各国统计学家的极大兴趣，并得到了广泛的应用。

脸谱图在应用上存在的最大问题是变量安排的次序，用哪个变量来画脸的哪个部位存在着人的主观性，而不同的用法给人留下的脸谱的印象大不相同，严重时可能会失真。研究表明，随机地安排变量大约会造成 25%的误差变动。所以在某个领域的实际应用过程中，脸谱图都要经过一定的探索才能绘制出合理的脸谱图，工作量比较大，并且不一定适合其他领域的应用。为了解决这个问题，可以采用主成分分析方法来解决对应变量的分配问题。经过主成分分析后，取 L 个最大的特征值对应的特征矢量作为新变量输入，L 的取值范围为 $k \leqslant L \leqslant d$，其中 d 为维数，k 为主成分数，一般取方差贡献率在 96%以上的主成分数目，$L=d$ 表示全部保留所有主成分，当 $d>$ 时 18，可以增加脸谱特征，也可以只保留 L（$L \leqslant 18$）个主成分。 但是当 L（$L \leqslant 18$）个主成分的方差累计贡献率低于 95%时，则只有增加脸谱特征。

脸谱图中数据的大小对图形有很大的影响，各变量的范围如表 14.3 所示。可以对原始数据进行归一化处理后再画脸谱。

表 14.3　脸谱图各变量的定义及范围

变　量	变量在脸谱上的定义	数值范围	变　量	变量在脸谱上的定义	数值范围
x_1	OP的长度	0～1	x_{10}	眼的位置（纵坐标）	0～1
x_2	x轴与 OP的角度	0～1	x_{11}	眼的位置（横坐标）	0～1
x_3	$OU=OL$ 的长度	0～1	x_{12}	眼的倾斜角	0～1
x_4	脸的上椭圆离心率	0.2～0.8	x_{13}	眼的椭圆离心率	0.4～0.8
x_5	脸的下椭圆离心率	0.2～0.8	x_{14}	眼的长轴的长度	0～1
x_6	鼻子的长度	0.1～0.7	x_{15}	眼球的位置	0～1
x_7	嘴的位置	0～1	x_{16}	眼到眉的高度	0～1
x_8	嘴的曲率	−5～5	x_{17}	眉的倾斜角	0～1
x_9	嘴的大小	0～1	x_{18}	眉的长度	0～1

请画出下列数据的脸谱图：

```
x=[0.13 0.04 0.55 0.25 0.15 0.1 0.1 3 0.4 0.3 0.6 0.2 0.3 0.7 0.5 0.3 0.2 0.5 0.3
   4 0.3 0.5]
```

解：

```
>> face(x);
```

得到如图 14.37 所示的脸谱图。

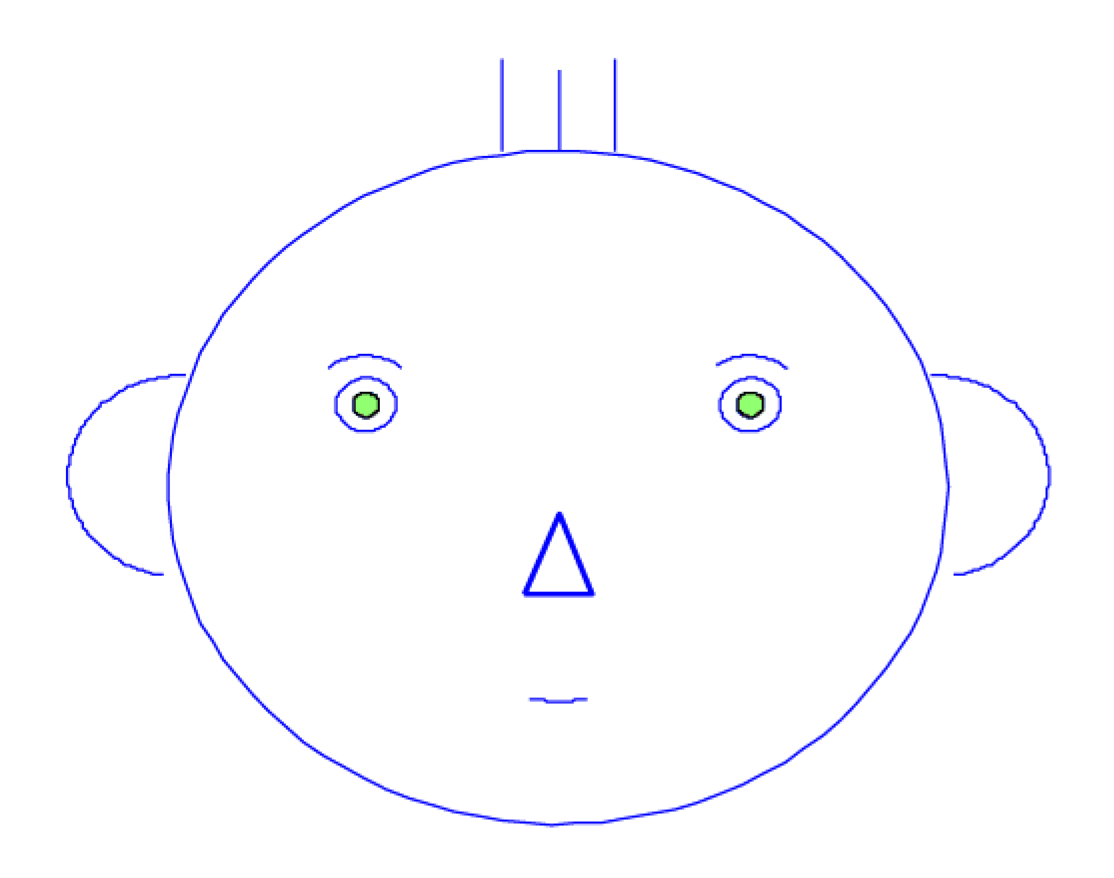


图 14.37　脸谱图

脸谱图也可以用其他软件绘制，图 14.38 即为用 R 语言画出的一个随机脸谱图。

随机脸谱图

1　2　3

4　5　6

7　8　9

图 14.38　R 语言画出的脸谱图

第 15 章

公式发现

15.1 公式发现概述

自然界存在着无数的规律（公式），除了已被发现的以外，还有很多规律需要人们去发现。在大量的工程问题中，同样也存在着大量的实验数据需要人们去寻找其内在的规律性。在这个过程中，计算机的广泛应用极大地提高了公式发现的效率，其中数据拟合是较为常用的方法。数据拟合是利用科学实验中得到的大量测量数据，去求自变量和因变量间的近似公式。但数据拟合虽然能解决一些实际问题，但它只能对一般实验数据找到满足精度的逼近公式。

随着人工智能技术的发展，机器发现技术得到发展。机器发现是指从一组观察结果或数据归纳中，找出这些数据的一个或多个规律。比较典型的系统有：科学发现系统 BACON，数据概念发现系统 AM 等。

给定一组可观察变量 $\boldsymbol{X}(x_1,x_2,\cdots,x_n)$以及这组变量的试验数据 $\boldsymbol{D}_i(d_{i1},d_{i2},\cdots,d_{in})$，$i=1,2,\cdots,m$，机器公式发现系统要找出该组变量满足的数学关系式，即对于任意一组试验数据均满足的关系式 $\boldsymbol{f}(x_1,x_2,\cdots,x_n)=c$，其中 c 为常数。

所找出的关系式是任何形式的数学公式，可分为以下几类。

（1）变量的初等运算：$f(x,y)=x\theta y$，其中 θ 为 +、−、×、／。

（2）变量的初等函数运算：$f(x)=c$，其中 $f(x)$为初等函数。

（3）初等函数的任意组合：$f(x,y)=a_1f(x)\theta a_2f(y)$，其中 θ 为 +、−、×、／。

（4）复合函数的运算 $g(f(x))=c$，其中 $g(x)$、$f(x)$均为初等函数。

（5）复合函数的任意组合 $h(a_1g_1(f(x))\theta a_2g_2(f(y)))$，其中，$h(x)$、$g(x)$、$f(x)$均为初等函数，$\theta$ 为 +、−、×、／。

（6）多个初等函数的组合：$f(x,y)=a_1f_1(x)\theta a_2f_2(x)\ldots\theta a_kf_k(y)$，其中 $f(x)$、$f(y)$均为初等函数，θ 为 +、−、×、／。

（7）分段函数：对于不连续的点，分别用不同的函数加以描述。

对于多变量更为复杂的公式的发现，一般是先寻找两变量的关系，再逐步扩充为多变量的关系。

15.2 公式发现系统中的知识

经验公式发现系统 FDD（formula discovery from data）的基本思想是利用人工智能启发式搜索函数原型寻找具有最佳线性逼近关系的函数原型，并结合曲线拟合技术及可视化技术来寻找数据间的规律，其总体结构如图 15.1 所示。

FDD 系统在搜索时，对某一变量取初等函数和另一个变量的初等函数或原始数据进行线性组合，即从原型库中选取逼近效果最好的少数几个初等函数作为基函数，并进一步形成组合函数，直至找到最后的目标函数。FDD 系统的启发式函数形式为：

$$f(x_1)=a+bf_1(x_1)$$

线性逼近误差公式为

$$dt=(a+bf(x_1)-f(x_2))/f(x_2)$$

在选择过程中，总是选取 dt 最小的函数作为继续搜索的当前节点。

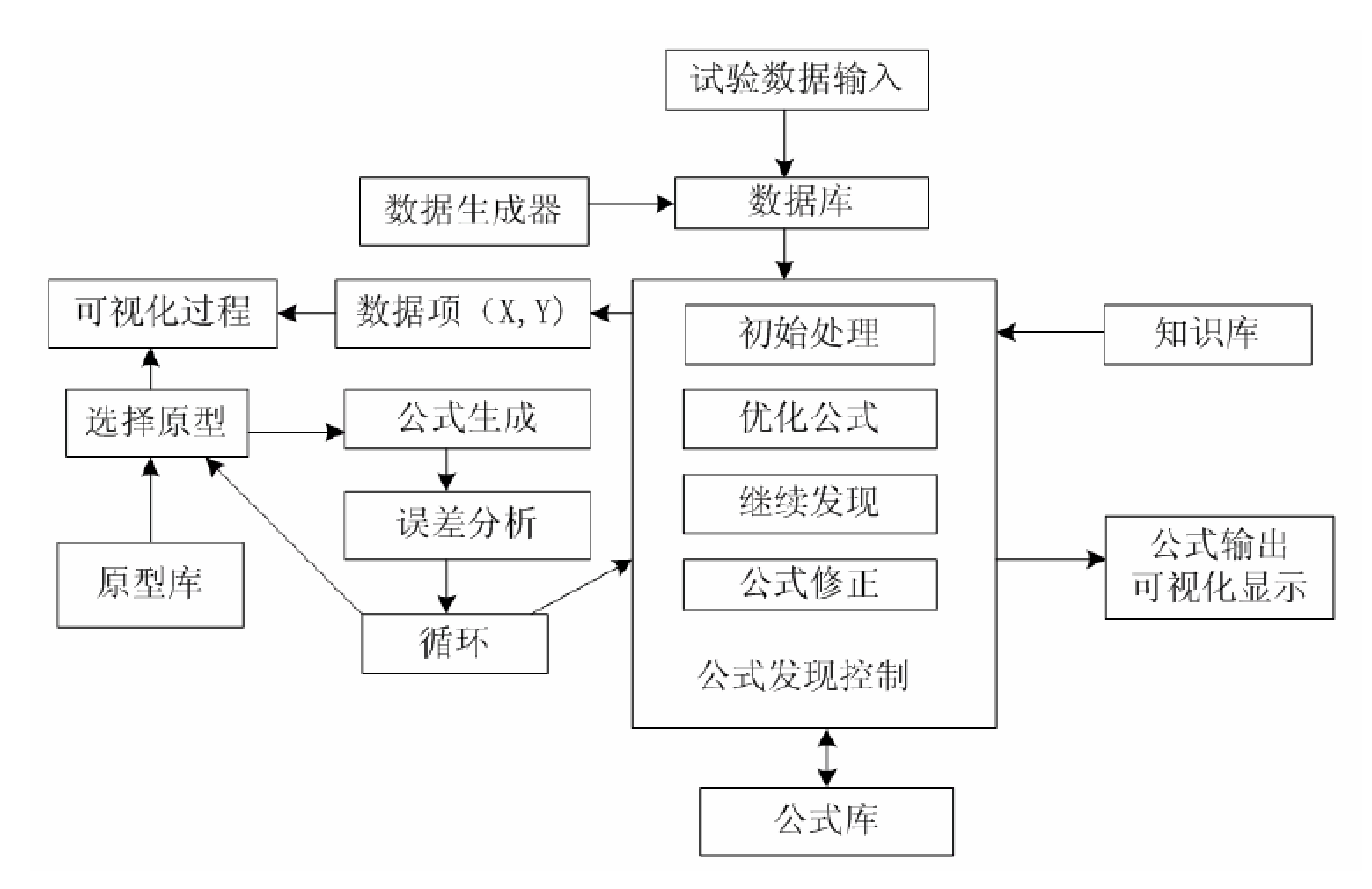


图 15.1　公式发现系统

具体过程如下：

步骤 1：固定变量 y，对变量 x 进行学习，即在现有原型基础上，根据原型库中的函数依次对实验数据进行匹配，用最小二乘法求出 a、b 系数，记录每次学习后 a、b 值和误差 dt，若某一原型经过线性组合后与试验数据的相对误差小于一给定值，则学习成功，求得 $f(x)=a+b*y$，否则转步骤 2。

步骤 2：固定步骤 1 所求得的 $f(x)$，对 y 进行学习，方法同步骤 1，求得 $f(x)=a+bf(y)$，若此时的误差小于给定值，则学习成功。否则继续搜索。此后的搜索可以是初等函数的线性组合，或嵌套函数形式。

15.2.1　规则一（函数规则）

主要的基本规则有以下几种。

1. 发现常数

若一个变量 x 取一个常数，则建立的该变量常数的公式为

$$x=c$$

2. 两变量的初等运算组合

当两变量进行初等运算时，若等于常数，则建立的该变量的初等运算关系式为

$$a_1x_1\theta a_2x_2$$

其中：θ 为 +、−、×、／。

3. 变量取初等函数

若某变量取初等函数等于常数，则建立的该变量的初等函数关系式为

$$f(x)=c$$

其中：$f(x)$为初等函数。

4. 两变量取初等函数的线性组合

两变量分别取初等函数后的线性组合等于常数，则建立的两变量取初等函数的线性组合关系式为

$$a_1f(x_1)+a_2f(x_2)$$

其中：$f(x_1)$、$f(x_2)$为初等函数。

5.某变量取某一初等函数与另一变量的线性组合

对某一变量 x_i 取初等函数后与另一变量 x_j 进行线性组合，若为常数，则建立的关系式为

$$c_1f(x_i)+a_2x_j=c$$

6. 对某一变量 x_j 取初等函数

另一变量 x_i 取两个 x_i 的初等函数进行线性组合，若为常数，则建立的关系式为

$$c_1f_1(x_i)+c_2f_2(x_i)+c_3g(x_j)=c$$

7. 建立新变量（启发式 1）

若两变量的某初等运算接近常数，则建立新变量为该两变量的某种初等运算。

8. 建立某变量的某种初等函数为新变量（启发式 2）

若某变量的某种初等函数与另一变量或它的初等函数进行线性组合接近常数，则建立该变量的初等函数为新变量。

以上规则的嵌套或递归使用，将形成变量的任意函数间的任意组合。在应用规则时，利用可视化技术将减少各种函数和各种运算的运算，大大节省搜索时间。

15.2.2 规则二（导数规则）

规则二是有关差分和差商的知识应用。

1. 差分发现常数

当某一变量差分 y 取一个常数 c，则建立的该变量等于常数的公式为

$$y=a+cx$$

2. 差商发现常数

当两个变量差商取一个常数 c，则建立的该变量等于常数的公式为

$$y'=c$$

3. 特殊函数形式导数函数

（1）阶差（向前差分）法判定类型

若 $\Delta^2 y_i$ =定值，则方程为：$y=a+bx+cx^2$

若 $\Delta^3 y_i$ =定值，则方程为：$y=a+bx+cx^2+dx^3$

若 $\Delta(y_i)^{-1}$ =定值，则方程为：$y^{-1}=a+bx$

若 $\Delta^2(y_i^2)$ =定值，则方程为：$y^2=a+bx+cx^2$

若 $\Delta^2(x_i / y_i)$ =定值，则方程为：$y=x/(a+bx+cx^2)$

若 Δy_i 成等比数列，则方程为：$y^2=ab^x+c$

若 $\Delta \log(y_i)$ 成等比数列，则方程为：$\log(y)=a+b^x+cx^2$

若 $\Delta^2 y_i$ 成等比数列，则方程为：$y=ab^x+cx+\mathrm{d}$

（2）差法判定类型

若 $\Delta \log(y_i) / \Delta \log(x_i)$ =定值，则方程为：$\log(y)=ax^{\mathrm{b}}$

若 $\Delta \log(y_i) / \Delta x_i$ =定值，则方程为：$y=ab^x$

若 $\Delta(x_i y_i) / \Delta x_i$ =定值，则方程为：$y=a+b/x$

若 $\Delta(x_i / y_i) / \Delta x_i$ =定值，则方程为：$y=x/(ax+b)$

若 $\Delta y_i / \Delta(x_i)^2$ =定值，则方程为为：$y=a+bx^2$

4. 两变量的导数运算组合

当两变量差分商后进行初等运算若等于常数，则建立的该变量的初等运算公式为

$$\Delta f(x_1)\theta f(x_2)=c$$

其中：θ 为＋、－、×、／，Δf 为差分或差商计算。

5. 两变量取导数运算的线性组合

两变量分别取导数运算后的线性组合等于常数 c，则建立的两变量取导数运算的线性组合关系式为

$$a_1\Delta f_1(x_1)+a_2\Delta f_2(x_2)=c$$

其中：$\Delta f_1(x_1)$、$\Delta f_2(x_2)$为导数运算。

以上规则和规则一嵌套或递归使用，将形成变量的任意函数和导数运算组合。

15.2.3　多维函数扩展规则

多维函数空间由初等函数、初等函数组合、复合函数、复合函数组合、函数导数等组成。初等函数组合是初等函数之间的运算组合；导数处理包括一阶差分、二阶差分、一阶差商、二阶差商等。

多维函数空间中的函数作用于变元或常数、函数仍然属于函数空间，这样为计算机对函数空间的处理提供了可以递归的前提。

1. 扩展到三维函数公式的启发式规则

设给定 n 组不同的数据 $\{x_1^{(k)},x_2^{(k)},x_3^{(k)}\},k=1,2,\cdots,n$ ，存在不同的函数 f_1, f_2, f_3, f_4 以及常数 C_1,C_2,B_1,B_2，有如下函数关系：

（1）如果在给定 x_3 的情况下得出 x_1 和 x_2 的方程为

$$f_1(x_1)=C_1f_2(x_2)+C_2$$

在固定 x_2 的情况下得出 x_1 和 x_3 的方程为

$$f_1(x_1)=B_1f_3(x_3)+B_2$$

则有如下的启发式公式

$$f_1(x_1)=C'_1f_3(x_3)f_2(x_2)+C'_2$$

$$f_1(x_1)=C'_1f_2(x_2)+C'_2f_3(x_3)+C'_3$$

（2）如果在固定 x_2 的情况下得出 x_1 和 x_3 的方程为

$$f_3(x_1)=\mathrm{B}_1f_4(x_3)+B_2$$

则有如下多个启发式公式

$$f_1(x_1)\theta f_3(x_1)=(C'_1f_2(x_2)+C'_2)\theta(B'_1f_4(x_3)+B'_2)$$

其中：θ 为＋、－、×、／等操作。或者

$$f_1(x_1)=g(x_1,x_2)+C'_1f_4(x_3)+C'_2f_3(x_2)+C'_3$$

g 函数的结构形式实质上是函数 f_2 和 f_3 的复合形式，由于 f_2 和 f_3 有系数项也有常数项，故 f_2 和 f_3 复合函数形式根据具体函数的不同有不同的合并方式，通常用一个公式的函数项去替换另外一个公式的系数和常数。

2. 扩展到四维函数公式的启发式规则

设在三维数据的基础上增加一维数据 x_4，如果得到公式

$$f_2(x_2)=C_1g(x_1,x_3)+C_2 \text{ 和 } f_2(x_2)=C_3f(x_4)+C_4$$

则有如下启发式公式

$$f_2(x_2)=C^*_1g(x_1,x_3)f_4(x_4)+C^*_2$$

$$f_2(x_2)=C^*_1g(x_1,x_3)+C'_2f_4(x_4)+C'_3$$

3. 多维函数的扩展

通过增加函数变量可以实现对多维函数变量公式的发现多维函数扩展规则给出了函数公式的具体框架表示形式，最后必须通过给定的数据对各个启发式公式进行检验，决定公式的取舍。

15.2.4 规则三

1. 函数规则

对某一变量取函数空间中的一个函数后与另一变量的函数进行线性组合，得到函数公式后，代入和值，取函数公式两边值的误差最小，则有函数公式

$$a_1 + b_1 * f(x_1) = a_2 + b_2 * f(x_2)$$

2. 函数嵌套规则

对函数规则嵌套或递归使用，将形成变量的任意组合。

3. 误差规则

（1）误差最小规则：选择误差最小的公式进入下一次迭代；

（2）误差收敛规则：保留误差减少的搜索方向，上一次迭代的误差大于目前的误差，则对于这一搜索方向予以保留。

4. 终止规则

终止规则由两部分组成：一是强制终止；二是自然终止。强制终止通过对算法参数的设定，主要是通过对迭代次数的设定完成终止准则。自然终止由两种情况组成：一种是找到一组满足给定误差的公式；另一种情况是判断出误差增大时，停止该路径的搜索。

15.3 基于 MATLAB 的公式发现

例 3.37　炼钢厂出钢时所用盛钢水的钢包，在使用过程中由于钢液及炉渣对包衬耐火材料的侵蚀，使其容积不断增大，钢包的容积与相应的使用次数（即包龄）的数据如表 15.1 所示。

表 15.1　钢包容积数据

使用次数 x	容 积 y	使用次数 x	容 积 y
2	106.42	11	110.59
3	108.20	14	110.60
4	109.58	15	110.90
5	109.50	16	110.76
7	110.00	18	111.00
8	109.93	19	111.20
10	110.49		

解：

根据公式发现搜索过程的原理，可编程计算得到如下结果。

```
>> x1=[2 3 4 5 7 8 10 11 14 15 16 18 19]';
>>y1=[106.42 108.20 109.58 109.50 110.00 109.93 110.49 110.59 110.60 110.90
      110.76 111.00 111.20]';
>> [f_F,d,x]=FDD(x1,y1);
>> f_F='1./x'         'log(x)'         %函数形式
>> x=4.7141          -0.0903          %回归系数
```

通过此方法搜索得到的函数要比纯粹的回归分析得到的函数表达式更具有物理意义，即更能反映变量间的关系。

在应用 FDD 时，应注意以下几点。

（1）要根据数据点间的关系，以及变量定义域等因素，选择适当的函数。在搜索过程中，既可以一次性选择函数，也可以多次选择。

（2）原型函数据库越丰富，就越利于公式的搜索。如果原型函数不能满足要求，可以自己添加其他的函数式。

（3）搜索时，一般要对两个方向（即 $x \to f(y) \to f(y) \to f(x)$ 以及 $y \to f(x) \to f(x) \to f(y)$）进行搜索，然后比较误差大小，最终确定函数形式。此例中两个方向搜索得到的函数关系就不相同。

例 3.38　对以下数据进行公式搜索。

```
x1=(2:21)';
y1=[21.656 9.24056 17.1851 23.4166 9.9625 14.9247 24.5738 11.1534 12.9081 24.9996
    12.786 11.2431 24.6274 14.7897 10.0221 23.5087 17.0400 9.27157 21.7803
    19.3598]';
```

解：

对数据点作图，得图15.2。

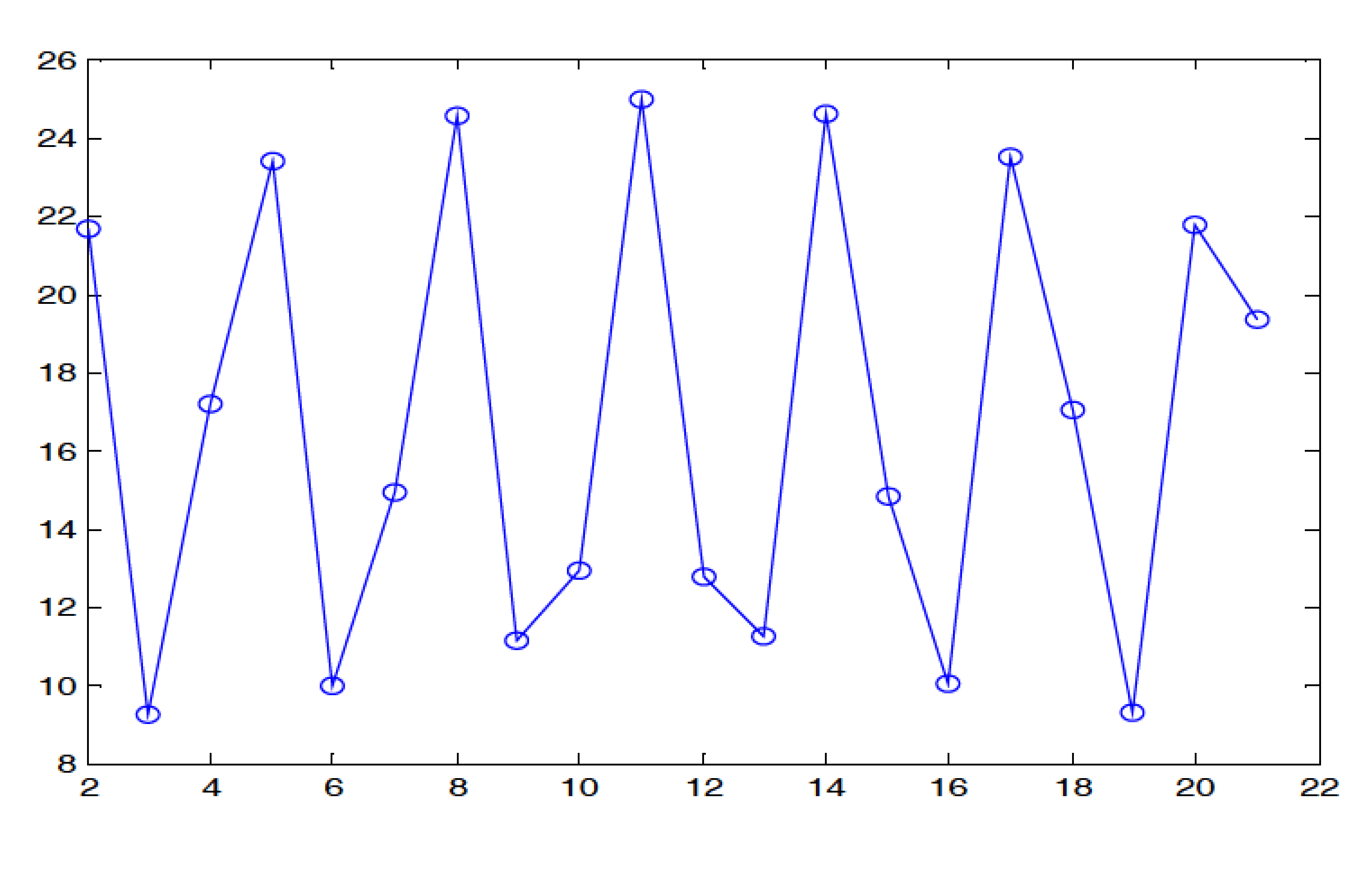


图 15.2　数据间的关系图

从图中可看出，变量之间具有周期性，所以除选择一般的函数外，可选择三角函数。

```
>> [f_F,d,x]=FDD(x1,y1);
>> f_F='(sin(x)).^2'    'sqrt(x)'          %函数形式
>> x=3.0001             1.9998             %回归系数
```

事实上，此例的数据点的关系是根据函数关系式所得出

$$\sqrt{y} = 2\sin^2(x) + 3$$

第 16 章

多媒体数据挖掘技术

16.1 多媒体数据挖掘技术概述

多媒体数据包括结构化的数据、半结构化的数据和非结构化的结构，如音频数据、视频数据、文本数据和图像数据等。

由于存储技术的迅速发展以及网络应用的普及，网络上已经拥有大量的文本和图像数据。此外，新闻服务每天也会产生大量的视频和音频数据。如何应用好这些多媒体资源的问题，已越来越引起人们的注意。这些研究以前主要集中在基于内容的信息检索方面，取得的成果在一定程度上解决了信息搜索和信息资源发现的问题。但因为信息检索只能获取与用户要求了相关的“信息”，不能发现和分析出蕴含在大量多媒体数据中有价值的“知识”。为此，更为迫切的是需要研究比多媒体信息检索更高层次的新方法，这就是多媒体数据挖掘。多媒体数据挖掘就是通过综合分析多媒体数据的内容和语义，从大量多媒体数据中发现隐含的、有效的、有价值的、可理解的模式，得出事件的发展趋向和关联关系，为用户提供问题求解层次上的决策支持能力。

现实世界上大量的数据以多媒体数据形式存在，目前绝大多数数据挖掘工具是针对关系数据库开发的，因此有必要对多媒体数据挖掘方法进行研究。多媒体挖掘的方法有两种，一种是先从多媒体数据库中提取结构化的数据，然后再用传统的数据挖掘工具在这些结构化的数据上进行挖掘；另一种解决办法是研究开发可以直接对多媒体数据进行挖掘的工具。严格地讲，多媒体数据是指由多种不同类型媒体数据组成的，包括文本、图形、图像、声音、视频、动画等不同类型的媒体数据，为了挖掘多媒体数据，必须对两种或多种类型的媒体数据进行综合挖掘。

16.1.1 数据类型

在描述多媒体数据时必然涉及一些多媒体特征。因此，需要有捕捉复杂数据类型和数据关系的方法。比如时间约束就包括“播放前”“播放后”等。假设有两个对象 A 和 B，A 包括 2000 帧，B 包括 3000 帧，A 所在的时间段是 4/95~8/95，B 的时间段是 5/95~10/95。需要有一定的数学模型恰当地描述这些特征。

恰当的数据模型对描述一个多媒体管理系统是至关重要的。可以用关系型、面向对象型及以对象—关系型数据模型来描述多媒体数据。关系模型能够捕捉数据之间的关系；面向对象模型可以描述复杂结构。图 16.1 所示的是面向对象的模型，图中的每个对象和数据模型中的每个对象相关联。对象的属性可以由实例变量描述，包括时间片、帧和内容描述等。在关系模型中，对象和一个关系的实例相关。在对象—关系模型中，实例的属性值可以是一个对象。例如对象 A 的实例属性值“时间片”就是成对出现的（4/95，8/95），图 16.2 所示的即为用对象模型描述对象 A。对于同样的对象，用图 16.3 对象—关系模型描述，可以支持多媒体数据的复杂关系。这些关系可以是对象之间的时序关系，例如“同时播放”“播放前”或“播放后”。

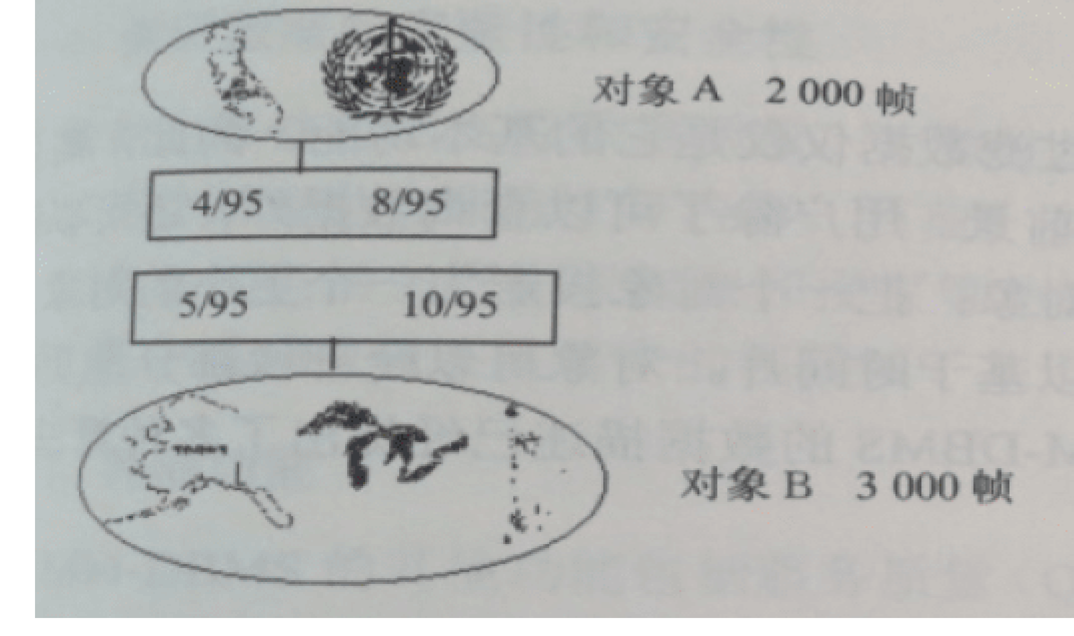


图 16.1　面向对象模型

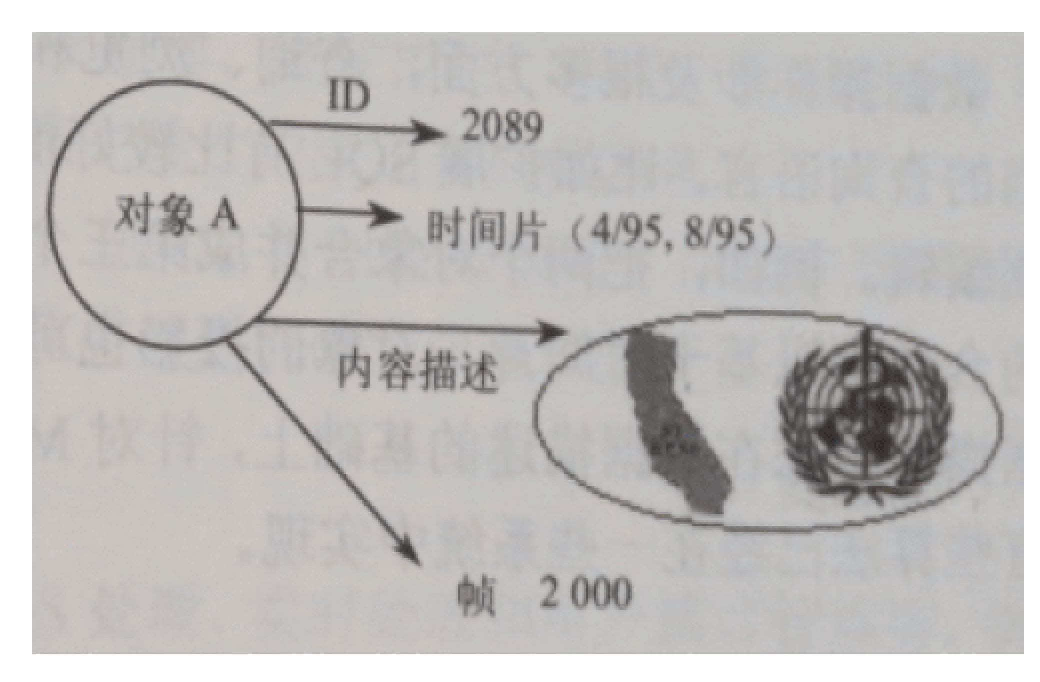
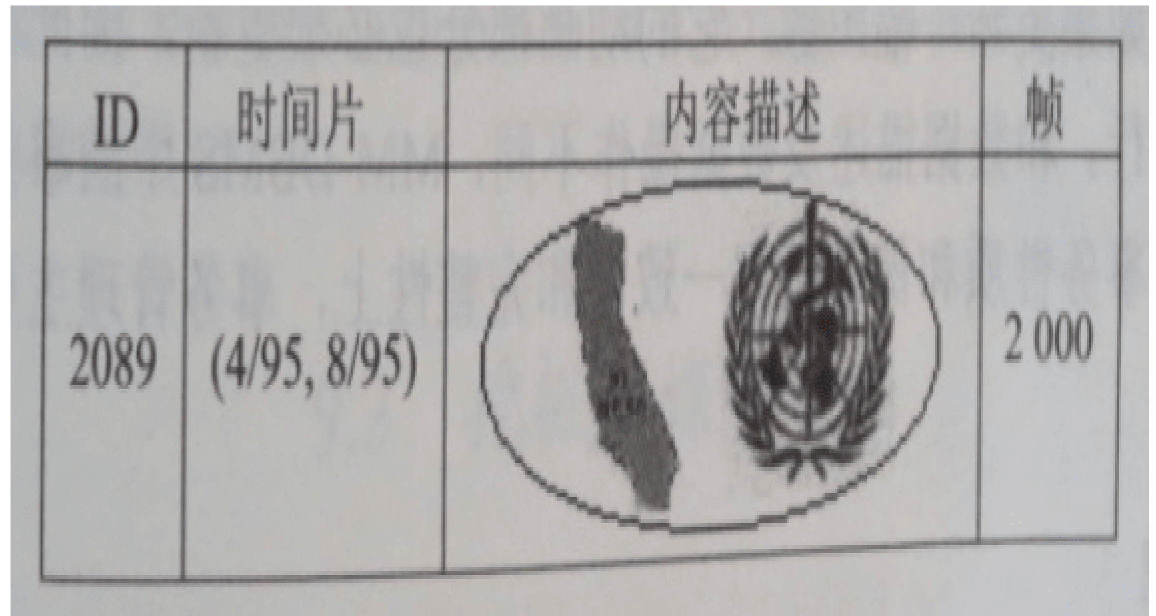


图 16.2　对象模型



ID	时间片	内容描述	帧
2089	(4/95, 8/95)		2 000

图 16.3　对象—关系模型

无论哪种模型都需要扩展才能捕获时间结构和其他特征。与数据模型相关的还有查询语言。查询语言应该具备操作多媒体数据库的能力。例如，查询播放视频片段中第 500~1000 帧等。

16.1.2　多媒体数据库管理系统（MM - DBMS）

多媒体数据库系统是多媒体挖掘的对象，它由多媒体数据库管理系统和多媒体数据库构成。其中，多媒体数据库用于存储和管理多媒体数据，多媒体数据库管理系统负责对多媒体数据库进行管理。多媒体数据库管理系统对存储、管理和检索多媒体数据提供支持。在某种意义上，多媒体数据库是一种异构数据库系统，因为它管理的文本、图像、视频和音频等数据的媒体各不相同。

多媒体数据库管理系统不但提供包括查询处理、更新处理、事务管理、存储管理、元数据管理、安全性以及完整性在内的典型数据库管理系统功能，而且要满足异构数据的特殊需要。例如，声音和图像必须同步播放，各种接口要求等问题。此外，实时处理也是它所面临的主要问题之一。

多媒体数据库管理系统必须支持基本的数据库管理系统功能，这些功能包括数据操作（查询、更新处理）、事务管理、元数据管理、存储管理、维护数据安全性及完整性。由于多媒体数据库管理系统面对结构化的和非结构化的数据，处理某些数据类型（如音频、视频等）非常困难，所以上述功能会变得很复杂。除了其基本功能外，管理系统还必须解决多媒体数据的实时处理和同步等问题。

1. 数据操作

数据操作涉及很多方面，查询、浏览和过滤数据仅仅是它的基本功能。需要更恰当的查询语言来实现数据操作，扩展 SQL 具有比较好的前景。用户除了可以查询数据外，还可以实现数据编辑。例如把两个对象合并成第三个对象；把一个对象投影为一个更小的对象；对象可以全部或部分地更新。数据操作是建立在数据描述的基础上，而对于后者已经提出了多种算法，其中有些算法已经在一些系统中实现。

2. 事务管理

系统中的事务管理是很重要的问题，因为在多数情况下，动画是和多媒体对象相关联的。例如，如果更新一幅图像，它的动画部分也必须更新，因此，这两个操作必须作为一个事务来执行。和数据描述及数据操作不同，系统中的事务管理仍是一个较新的领域，在维护事务性质和保证数据一致性和完整性上，事务管理主要使用并发挖掘和恢复机制。

3. 元数据管理

对音频、视频数据的描述需要大量的元数据。就视频而言，可能需要多种帧的信息，这些信息通常保存在元数据中。

元数据在模式匹配（识别）中起着关键作用。为对多媒体数据进行数据分析，必须了解用户想要查找的内容。例如，在视频剪辑中，为了识别多种模式，就必须事先存储一些模式才能使模式识别顺利进行，这些模式信息以元数据的形式出现。

互联网技术的快速发展使元数据管理变得更加复杂，也使得元数据管理更富有挑战性。

4. 存储管理

存储管理主要包括设计和开发适合多媒体需要的、特殊的索引方法和存取策略。虽然基于内容的数据存取在多媒体应用中占有非常重要的地位，但现在还没有一种高效的基于内容的多媒体数据存放方法。存储管理的另一个问题是数据缓冲问题，需要提高多媒体数据在高速缓冲中的命中率。与结构化数据相比，需要研究多媒体数据在使用高速缓冲时的特殊性和特殊算法。此外，存储技术还需要具有集成不同类型数据的能力，例如，一个多媒体数据库系统可以包含视频、音频和文本数据，不应仅仅由一种数据类型构成，这些不同数据类型还涉及同步问题，需要有恰当的存储机制来实现异构数据的连续存取。

5. 保证数据的完整性和安全性

数据完整性包括支持数据质量、完整性约束处理、并发挖掘、多用户数据更新、数据恢复以及数据的准确性等内容。目前，实现完整性约束还有很多困难。

安全机制包括支持存取权限和授权等功能。例如，针对视频数据，存取控制规则应该于整个视频剪辑还是单个的视频帧。

6. 其他功能

多媒体数据库管理系统的其他功能包括服务质量（实时处理和用户接口管理等）。例如，在某些情况下，可能需要连续显示数据；有时又需要支持服务中断容错；为有效地实现多媒体数据的输入和输出，必须提供恰当的多媒体数据接口等。

16.2 文本挖掘

文本数据不同于关系数据。在大多数情况下，文本数据是非结构化的，有些情况下它是半结构化的。例如，一篇文章是半结构化的，文章有标题、作者、摘要和段落。段落是非结构化的，而其格式是结构化的。

经过数十年的发展，信息检索系统和文本处理系统有了长足发展。例如，只要根据给定的属性值和关键字就可以检索出文档，一些文本处理系统还可以检索出文本之间的关联关系。

文本挖掘是从非结构化的文本中发现潜在的概念以及概念间的相互关系，它从大型文本数据库提取尚未被人们所认识到的模式或关联。有的信息检索和文本处理系统可以发现字词和段落之

间的关联关系，因此也可以看成是文本挖掘系统。

很多数据挖掘的工具和技术都是针对关系数据库的，针对文本数据库的挖掘工具较少，因此当前的数据挖掘工具不能直接应用于文本数据。

挖掘非结构化数据的方法如下：

（1）使用特征标记技术从非结构化的数据库提取数据和元数据，并把提取的数据存入结构化的数据库中。应用现有的数据挖掘工具在结构化数据库中进行挖掘，如图 16.4 所示。

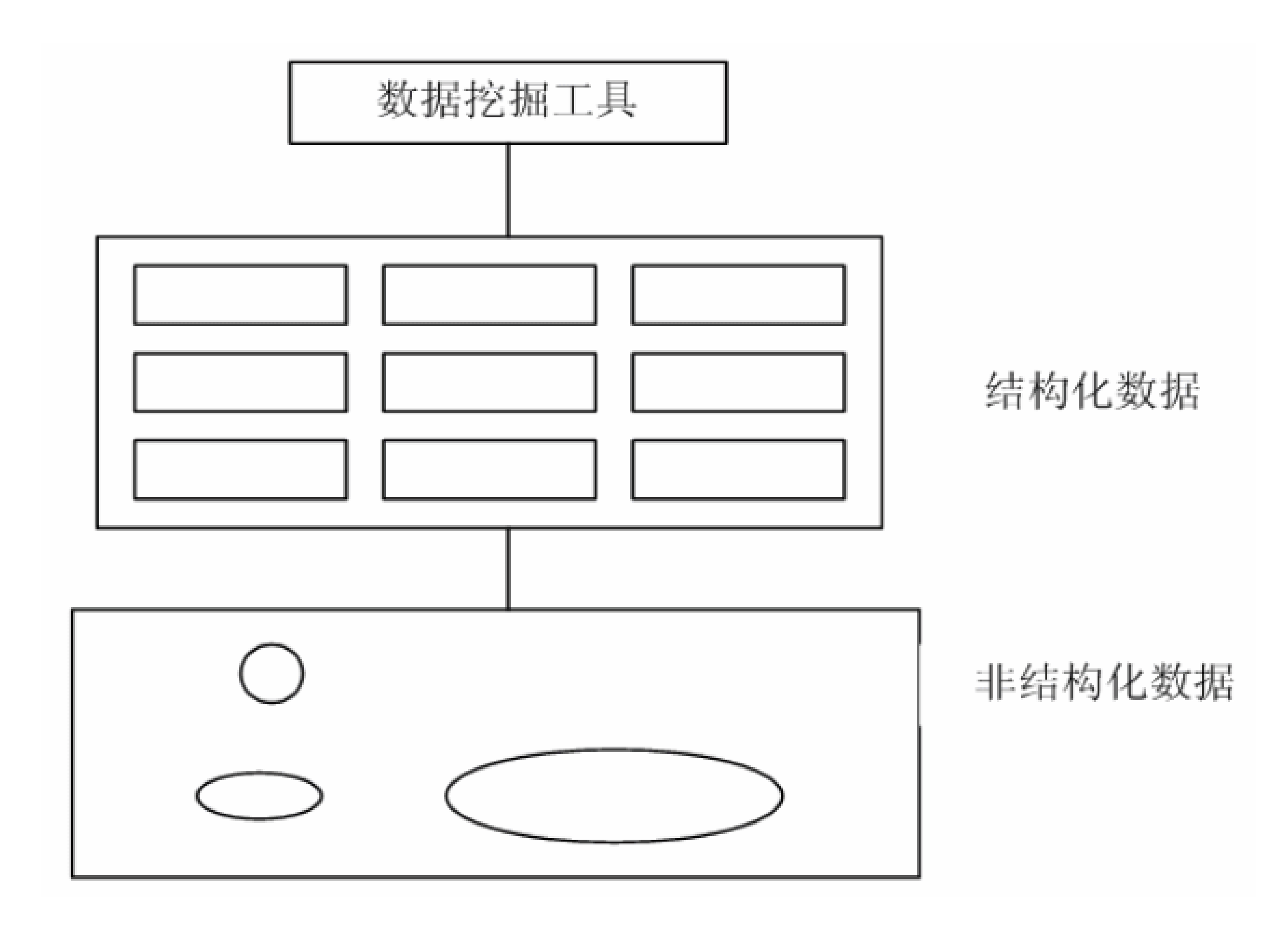


图 16.4　转换为结构化后挖掘

（2）将数据挖掘工具和信息检索工具集成在一起，目的是开发适合非结构化数据库的数据挖掘工具，如图 16.5 所示。

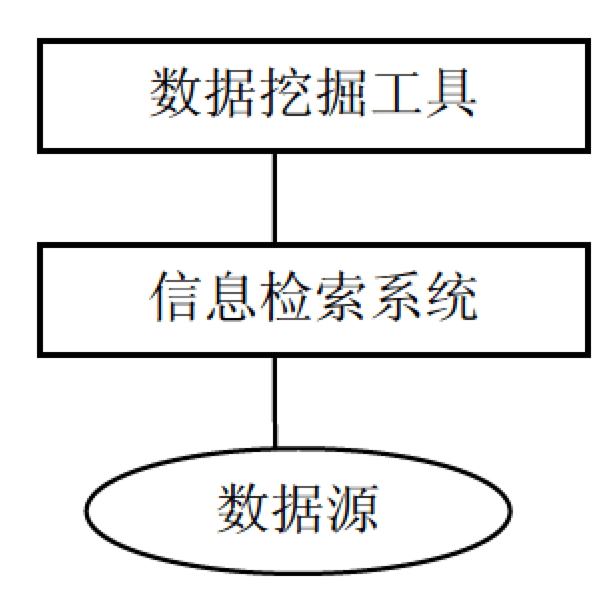


图 16.5　增强的信息检索系统

（3）开发直接应用于非结构化数据库的数据挖掘工具，如图 16.6 所示。

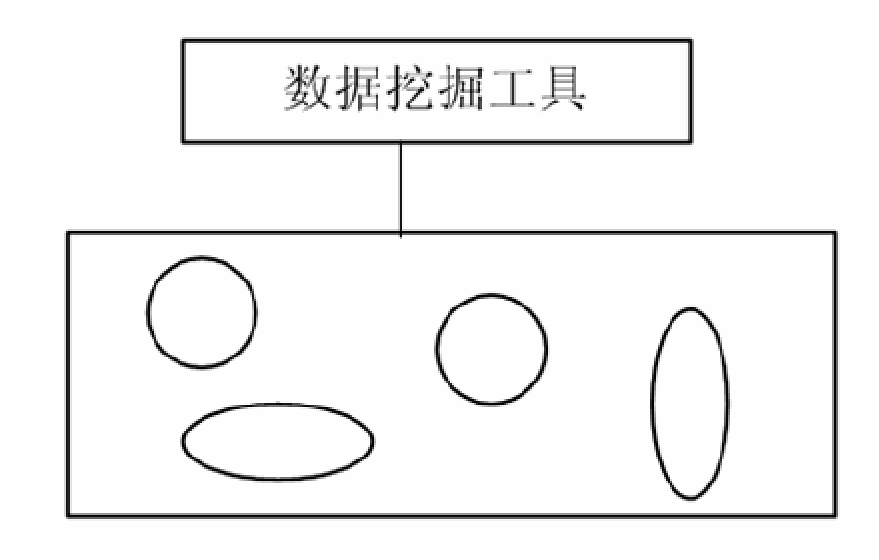


图 16.6　非结构化数据的直接挖掘

把文本数据转换并存入关系数据库时，必须防止关键信息的丢失。对转换后的数据库进行挖掘前，需生成数据仓库。这个数据仓库的实质是关系数据库，数据来源于文本数据中的重要数据。也就是说，必须有一个转换器，输入数据是文本集，输出是包含文本中关键字的表。

挖掘文本数据库的方法有两类：基于关键字的关联分析和文本分类分析。

16.2.1 基于关键字的关联分析

首先收集经常一起使用的关键词或词汇，然后找出其关联或相互关系。与文本数据库大多数数据分析和搜索引擎中的方法一样，关联分析首先要对文本数据进行分析、词根处理（即词根还原，一个词的多种变形视为一个词，如 do、done、doing、does、did 均视为一个词），去除停用词（对文章词义分析无意义的词，主要是英文中介词、冠词及中文中的虚词，如 in、the、of 等），然后调用关联关系挖掘算法。在文本数据库中，把每个文档作为一个事务，文档中的关键词组可视为事务中的一组事务项，这样文档数据库关键字关联规则挖掘的问题就转化为事务数据库中项集的关联规则挖掘问题。

16.2.2 文档分类分析

文档分类分析是一种重要的文本挖掘方法。通过对文档分类分析，把大量的联机文档自动分类组织，便于对文档的检索和分析。其过程通常包括如下几个主要阶段：文本预处理、文档的表示、维数约简、分类器的学习、分类器的测试以及性能评价。这个过程是一个反反复复，不断调整和反馈的过程。有些情况下为了研究的需要研究人员会自己建立文档集，更多的情况是使用国际上流行的、已经建立好的权威的文档集。

1. 文档预处理

在文本分类中，训练集的选择至关重要，选择的原则是国内外使用广泛、权威和规范的语料库，这样使得分类结果具有可比性，同时也便于认真细致地分析结果和算法的优劣。在英文语料库中，已经有受到国内外认可和广泛使用的路透社语料库（Reuters-21578、RCV1 和 RCV2）、TREC 文档集、20NewsGroups 和 OHSUMED）等文档集。而对于中文语料库，现有的语料库有复旦大学中文文本分类语料库）、北京大学的 Web 测试集等。这些文档集或多或少存在一些缺陷：有些存储格式不尽相同，一般不能直接使用；有些文档可能不完整，存在一些不规范字符；有些文档集存在不少的重复文档；有些文档集中的文档直接从网上下载，内容复杂，格式不规范，并且编码格式多样。这些问题严重影响文本分类系统后续的工作以及分类性能，所以必须进行一些前期的数据预处理工作，去除文档集中的噪音信息、将其内容规范化，使得文档符合分类模型的输入要求。

（1）去除格式标记。

文档中的格式标记去除是指去除语料库中的一些格式，提取文档里的部分内容，转换为文本分类系统需要处理的格式和内容。例如，一般只关心文档的标题、正文和超链接描述，处理时就可以通过 <TITLE> 、<BODY> 和 <A> 等标签提取相应文档内容。

（2）去除停用词和词干化。

停用词是指语言中的功能词，这些词在文档中出现次数很多而本身没有实际意义，中文中一般

称为虚词，英文中为冠词、代名词、助动词、介系词和连接词等。语料库中出现频率很低（一般为 1~3 次）的稀有词，也可以考虑去除，它们的数量通常很多，计算机难以处理，一般需要去除。

词干化主要是指去除英文中的前缀、后缀，保留单词中的词干部分。英文单词常由前缀、词根、后缀等部分组成，在句子中，单词还有性、数、格以及引起的词形变化，如 write、writes、wrote 和 writing 等，这几个单词就可以认为是表述同一个概念的。经过词干化处理后，就可以提取出代表这四个单词的共同词干：write。这样处理的目的也是便于计算机处理，减少文本处理中的特征维数。

词干化处理常采用自动机的规则方法，即将词形变化的规律总结成规则，然后通过自动机的方法对词形进行转换，转换过程当中可使用或者不使用词典。目前使用最广泛的词干化处理算法是 Martin Porter 提出的 Porter Stemmer 算法。

（3）中文分词。

分词是中文、日文等亚洲语言的处理中遇到的特殊问题。中文文本中词与词之间没有明确的分隔标记，而是连续的汉字串，因此，将汉字串切分为正确的词串的汉语分词问题无疑是实现中文信息处理的各项任务的首要问题，也是中文信息处理的基础。汉字的简体/繁体转换、信息检索和信息摘录、自然语言理解、文本分类、机器翻译、文本校对等中文信息处理系统同样首先需要将分词作为最基本的模块。

中文分词方法大致有三类：

- 机械分词方法：基于分词词表，按照字符串匹配的原理进行。根据切字串的方向，模式匹配又分为正向匹配法和逆向匹配法。根据每次匹配时优先考虑长词还是优先考虑短词，此方法又可分为最大匹配法和最小匹配法。
- 基于统计分词方法：先切分出与词表匹配的所有可能的词，这种切分方法称为“全切分”，然后运用统计语言模型和决策算法决定最优的切分结果。
- 基于规则和基于统计相结合的方法：首先运用最大匹配法作为一种初步切分；接着对切分的边界处进行歧义探测，发现歧义；再运用统计和规则相结合的方法来判别正确的切分，运用不同的规则解决人名、地名、机构名识别，运用词法结构规则生成复合词和衍生词。

2. 文档的表示

对文档进行预处理后，需要根据文本分类模型对文档进行相应的特征表示。一般文档的特征项应具有以下特点：特征项是能够对文档进行充分表示的语言单位；文档在特征项集合上的分布具有较为明显的统计规律；特征项分离比较容易实现，计算复杂度不太大。在文本分类中，按照文档特征的粒度来划分，常用的特征单位有词、词组、N-Gram（N 元）项和概念等。

（1）词。

在信息检索领域，词是使用最为普遍的文档特征。英文等西方语言中的词较易获取，而对于东方语言，则需要分词来得到词。对于中文等语言，也可以采用单个字来表示文档特征。

（2）N-gram。

N-gram 项一般是由相邻的 N 个词组成，经常在统计语言模型中使用。使用较多的是 Unigram（一元）、Bigram（二元）、Trigram（三元）。对于中文来说，N-gram 项一般由相邻的字构成，例

如：从“江西财经大学”中提取 2-gram 项，可以得到“江西”“西财”“经大”和“大学”4 个 2-gram 项。对于英文来说，N-gram 项既可以由相邻单词构成，也可以由相邻字母构成。

N-gram 项作为文档的特征，可以避免庞大的词典和复杂的分词程序。一般情况下，使用同样的分类方法，基于词的文本分类效果并不比基于 N-gram 项的好。在特征数目较小的情况下，基于 N-gram 项的文本分类效果甚至优于基于词的。但是，N-gram 项的语义显然没有真正的词那么明显，而且随着 N 的增大，N-gram 项的数目会呈指数增长，使算法的时间和空间消耗大大增加，所以 N 的取值一般不宜过大，目前取值一般不超过 3。

（3）词组。

词的文档表示法的一个显著缺点是原始文档中的大量语义信息被丢失了。例如段落、句子、词序和词性等都被忽略了。结果是虽然满足了计算机学习算法所需要的连续性，却打乱了人们正常的思维连续性。词组表示法的一个目标就是为了尽量挽回一些词语表示所滤去的有用信息。但词组表示法的表达能力并明显优于词的表示，因为词组降低了特征向量的统计质量，使得特征向量变得更加稀疏，让机器学习算法难以从中提取用于分类的统计特性。

（4）概念。

概念相比词语而言，具有更高的抽象性。在文本分类中，存在着一词多义和多词一义现象。此时，采用概念作为文本特征有诸多优点：首先大大降低分类空间的维数，从而节省了分类器的训练时间和分类期间用于相似比较的时间，时间效率大大提高；其次，可以避免一个重要的分类特征因为采用关键词的分散而削弱其分类的权重；再次，可以避免只采用关键词作为特征所产生的特征歧义，即虽然都采用了同一个关键词，但所代表的意义完全不同，从而提高分类的准确性。最后，基于关键词的分类假设关键词之间是独立的，但关键词之间不但存在同义、多义关系，还存在相关关系、相斥关系；将关键词映射到概念空间可以在一定程度上消除这种相关性。

定义了文档的特征后，就可以采用适当的文档表示模型进行表示。文档的表示模型是文本分类的基础，决定着文档表示为计算机容易存储格式的方法，会对分类任务产生较大的影响。为了处理的方便，通常的文本表示方法大都采用贝叶斯假设，即把组成文本的字或词对确定文本类别的作用认为是相互独立的，这样可以直接用文档中出现的字或词的集合代替文档。

文档表示模型有 4 个传统的模型，即布尔模型、向量空间模型、概率模型和逻辑模型，其中最为著名的就是向量空间模型。

（1）向量空间模型。

向量空间模型也称词袋表示方法。一个词袋是一个集合，它允许元素的重复，这样不但考虑词出现与否，而且考虑了词出现的频率。

向量空间模型的定义如下。

语料库中所有的词组成词表，一篇文档表示为向量空间中的一个向量，也即一个“袋子”：

$$\phi : d \mapsto \phi(d) = (tf(t_1,d), tf(t_2,d), \cdots, tf(t_n,d)) \in R^N$$

其中：$tf(t_i,d)$ 表示词 t_i 出现在文档 d 中的频率；t_i 为词表中的一个词；N 为词表的大小。这样，一个文档就映射到一个 N 维的空间，通常 N 是一个很大的数，而向量中很多元素为 0，即“数据稀疏”。

（2）特征权重表示方法。

在使用向量空间模型表示文档后，一般出于某种考虑，通过提高或降低某些特征的影响来进行特征权重的调整。权重的调整都基于以下两种考虑：一个词在某篇文档中次数越多，则对识别文档的贡献越大；一个词在不同文档中出现的次数越多，则它区分不同分档的能力越弱。

① 布尔权重。

布尔权重是最简单的一种权重表示方式，也称二值权值或二元权重。如果文档中出现了该词，那么在文档向量中该词所在位置的值为 1，否则为 0。

$$a_{ij}=\begin{cases}1,\ \text{如果}\ tf_{ij}>0\\0,\ \text{否则}\end{cases}$$

式中：tf_{ij} 为词 j 在文档 i 中的出现频率。

② 词频权重。

用词的频率作为权重是一种简单常见的表示方法，它直观且容易理解。

$$a_{ij}=tf_{ij}$$

其基本思想是某个特征在文档中出现的次数越多，它就越重要，但是文档一个高频词并不一定重要。

③ tf-idf 权重。

tf-idf 权重是一种使用非常广泛的权重表示方法，它考虑了词的文档频率信息。tf-idf 权重以词的逆向文档频数对词频作加权处理，其基本思想是词在文档中出现的次数越多就越重要；同时也认为词的文档频率越低，该词的重要性就越低。

$$a_{ij}=tf_{ij}\times\log\frac{N}{n_j}$$

式中：N 为文档集的文档总数；n_j 为词的文档频数。

当 N=n_j时，上述权重为 0，在小数据集上经常会发生这种情况，为防止出现这种情况，一般要做平滑处理

$$a_{ij}=\log(tf_{ij}+0.1)\times\log\left(\frac{N+1.0}{n_j}\right)$$

④ tfc 权重。

tf-idf 权重没有考虑文档长度对词权重的影响。为消除这种影响，tfc 权重对 tf-idf 权重作“归一化”处理，使每个文本的特征权向量都变成长度为 1 的单位向量

$$a_{ij}=\frac{tf_{ij}\times\log\left(\dfrac{N}{n_j}\right)}{\sqrt{\sum_{p=1}^{M}tf_{ip}\times\log\left(\dfrac{N}{n_p}\right)}}$$

式中：M 为文档集的词总数。

⑤ ltc 权重。

ltc 权重是 tf-idf 权重的一种变形形式，其表达式为

$$a_{ij}=\frac{\log(tf_{ij}+1.0)\times\log\left(\frac{N}{n_j}\right)}{\sqrt{\sum_{p=1}^{M}\left[\log(tf_{ip}+1.0)\times\log\left(\frac{N}{n_p}\right)\right]^2}}$$

或

$$a_{ij}=\frac{\log(tf_{ij}+1.0)\times\log\left(\frac{N}{n_j}+1.0\right)}{\sqrt{\sum_{p=1}^{M}\left[\log(tf_{ip}+1.0)\times\log\left(\frac{N}{n_p}+1.0\right)\right]^2}}$$

⑥ 熵权重。

熵加权法是基于信息论的加权算法，相对较为复杂，其表达形式为

$$a_{ij}=\log(tf_{ij}+1.0)\left\{1+\frac{1}{\log(N)}\sum_{p=1}^{M}\left[\frac{tf_{ip}}{n_p}\log\left(\frac{tf_{ip}}{n_p}\right)\right]\right\}$$

其中：$\frac{1}{\log(N)}\sum_{p=1}^{M}\left[\frac{tf_{ip}}{n_p}\log\left(\frac{tf_{ip}}{n_p}\right)\right]$ 表示词的平均不确定度或熵。

3. 常用文本分类模型

文本分类器是文本分类系统中的核心部分。目前，许多统计学习、机器学习和算法都在文本分类中得到了广泛的应用，基于统计学习、机器学习的文本分类技术已经成为主流技术。现已提出了许多文本分类算法，常见的有：最小二乘回归模型、k-近邻、决策树、朴素贝叶斯、神经网络、支持向量机、最大熵模型、Rocchio 分类器、关联规则和组合分类器等。这些算法的具体原理可参见相关参考书或本书的相关章节。

4. 文本分类器学习、测试和评价

文本分类器建立以后，需要进行分类器的学习训练过程，训练分类器的一些参数，然后对这些参数进行微调，最后评价它的分类。分类器性能的评估与比较是一个比较复杂的问题，目前尚未得到很好的解决。影响文本分类器实际分类效果的因素有很多，如语料库的选择、文档的表示、性能评估指标的确定、实验数据的分析与处理等。一般做法是：选用使用广泛的、规范的和权威的语实料库；选用适当的评价指标，目前常用的有精度、召回率和 F1 值等；对实验结果进行统计分析，如 T-检验等。

（1）文本分类器的学习和测试。

一般情况下，把原始的语料库分为训练集和测试集，它们的大小并不一定相等。测试集是为了微调分类器的参数，使分类器的性能较好。测试有封闭测试和开放测试。封闭测试时，测试集是训练集的一部分；开放测试时，测试集与训练集是独立同分布的两个数据集。因为封闭测试不具有可比性，文本分类中主要采有开放测试。对于语料库划分的问题，比较权威的是训练集为70%、测试集为30%。有的把上述方法所述的训练集再分为两部分，即语料库分为训练集、验证集和测试集三部分，验证集的目的是更加有效地微调参数和优化参数，这种方法有时也称保持法，也有的使用 k 折交叉验证法。因为计算机性能限制，在大规模文档集上经常使用这种方法。

（2）阈值策略。

在测试一篇新文档时，需要根据分类器给它的评分确定属于哪些类别以进行性能评价。因为类别之间并不是相互独立的，一篇文档可能属于这个类，也可能属于别的类。因此，需要对每个类别确定阈值，当文档大于某一类别的阈值时，就将文档划分到该类别中。确定这个阈值的过程，称为文本分类中的阈值策略。阈值策略的好坏会影响分类器的性能评价，目前在理论上没有一个好的解决方法，大多数是依靠实验和经验选取合适的策略。常用的阈值策略有排序阈值法、比例阈值法和局部最优评分法等。

（3）评价指标。

在文本分类器完成了训练和测试后，一个很重要的问题就是进行分类性能评估。要选择合适的评价指标评估一个算法的优劣，并且和算法进行性能比较。

设 a:正例测试文档被正确分类为属于该类的数量；

b:负例测试文档被错误分类为属于该类的数量；

c:正例测试文档被错误分类为不属于该类的数量；

d:负例测试文档被正确分类为不属于该类的数量。

可以用以下指标对分类效果进行评价。

精确率：它是分类系统结果与人工分类结果一致的文档在被分文档中的比率

$$\mathrm{Precision} = \frac{a}{a+b}$$

召回率：它是指人工分类结果应有的文档与分类系统一致的文档所占的比率

$$\mathrm{Recall} = \frac{a}{a+c}$$

其他定义

$$\mathrm{fallout} = \frac{b}{b+d}$$

$$\mathrm{accuracy} = \frac{a+b}{a+b+c+d}$$

$$\mathrm{error} = \frac{b+c}{a+b+c+d}$$

另外，常用的有 F1 测试值，它综合考虑精确率和召回率，也称为综合分类率，计算公式如下

$$F1=\frac{2\times \text{Precision}\times \text{Recall}}{\text{Precision}+\text{Recall}}$$

为了综合多个类别的分类情况，根据计算方式的不同，F1 值分为微平均 F1 值和宏平均 F1 值。前者的计算方式是首先需要在整个文档集内分别计算精确率和召回率的值，然后根据 F1 的计算公式计算微平均 F1 值。后者则是先计算每个类别的 F1 值，然后计算它们的平均值，即得宏平均 F1 值。很明显微平均 F1 值平等考虑每一个文档，因此它的值将主要受常见类的影响；而宏平均 F1 值平等对待每一个类别，因此它的得分主要受稀有类的影响。

Break-even 点也是一种常用的评价指标，对分类系统来说，精确率和召回率是相互相等的，提高其中任何一个都会引起另外一个指标的下降，一种做法是选取两者相等值来表征系统性能，这个值称作平衡点 BEP 值。当然，有时通过测试不能得到两者相等的值，这时取它们最接近的平均值作为 BEP 值，称为插值 BEP。

16.3 图像挖掘

图像内容包括地图、地质结构、生物结构等。图像处理涉及的研究领域有：检测模式的异常分析、基于内容图像检索和模式匹配等，其研究工作已有很长历史，研究成果已经应用到了许多领域，如美国航空航天局对空间图像和地质图像的挖掘，在医疗图像上的挖掘，在台风图像上的挖掘等。

图像检索是指图像数据相似检索，主要考虑两种多媒体标引和检索系统：①基于描述的检索系统，主要是在图像描述上建立标引和执行对象检索，如关键字、标题、尺寸和创建时间等；②基于内容的检索系统，它支持基于图像内容的检索，如颜色构成、纹理、形状、对象和对象之间的空间关系等。前者手工操作十分困难，自动完成的检索结果通常较差；而后者使用视觉的特征标引图像并按特征相似度检索对象，可以满足更多的系统需要。所以越来越多的系统采用后者图像检索技术。

在基于内容的图像检索系统中，通常有两种查询方式：基于图像样本的查询和基于图像特征描述的查询。基于图像样本的查询是指找出所有与给定图像样本相似的图像。具体过程是：通过索引从图像数据库中提取特征向量，与样本中提取的特征向量进行比较，可以检索出与样本图像相似的图像。图像特征描述查询要求给出查询图像的特征描述，系统把特征描述转换为特征向量，在数据库中检索与该特征向量相匹配的图像。如可以分别根据颜色、纹理或形状查询，也可以根据两个或三个参数特征进行综合查询。

图像处理主要是检测异常模式和图像检索；图像挖掘是发现所有异常的模式。因此，图像挖掘可以理解为从大型图像数据库中寻找不同图像之间的关联关系。

通过大量的研究可以发现直接对图像挖掘也是可能的，问题是确定何种挖掘结果最合适。图像的挖掘结果可以是关联规则、聚类图像、分类图像，也可以是检测异常模式。例如，通过开发生成图像之间规则的模板，再应用数据挖掘工具从中发现异常模式，即可完成图像中是否存在异常事物的判断过程。

检测异常模式并不是图像挖掘的结果，而仅仅是开始。图像挖掘需要研究现有数据技术能否应用在图像的分类、聚类和关联规则上。

16.4　视频挖掘

挖掘视频数据比挖掘图像数据更困难、更复杂。视频可以看作是移动的图像或动画。视频数据中包含丰富的内容线索。除图像具有视觉特征和空间特性外。视频数据还具有时间特性、视频对象特性、运动特性等。视频挖掘技术可以广泛应用于新闻视频、监控视频、记录影片、数字视频图书馆等信息挖掘。例如，从交通监视视频中分析出交通拥堵的趋势；从连续的侦察图像和视频新闻中分析出军队调动的动向；对广告的分析和挖掘；从国际视频新闻中挖掘出事件的关联、危机和灾害事件（水灾、火灾、疾病等）的发生模式等。

很多研究领域都涉及视频数据管理问题。例如，开发视频数据库的查询和检索技术，需要研究视频索引、查询语言和优化策略等。人们普遍认为，成功的视频挖掘系统首先要有一个成功的图像挖掘系统。

为了实现视频数据库中的模式匹配，用户应该预先定义好图像，然后用视频数据与这些图像进行匹配。可以认为视频挖掘就是从大型数据库中发现视频事件的关联和隐含模式，即通过综合分析视频数据的视听特性、时间结构、事件关系和语义信息，发现隐含的、有价值的、可理解的视频模式，得出视频表示事件的趋向和关联关系，提高视频信息管理的智能程度。

此外，还可以对视频结构进行分析和挖掘。挖掘视频的结构模型，称为镜头语法。镜头语法描述视频故事单元的构造模式，例如，一段新闻单元的构造模式可能是播音主持人后接说明场景，或是播音主持人与被采访对象镜头的交替对话模式等。

和文本挖掘、图像挖掘不同，迄今为止还没有真正意义上的视频挖掘研究成果。鉴于视频数据的特殊性，视频挖掘的研究范围还很有限，下面仅对几类典型的挖掘技术进行探讨。

16.4.1　结构挖掘

视频是非结构化的数据，提取视频结构是视频分析、视频索引、视频存取的基础。在视频特征提取时，仅按时间顺序把视频分割成单元镜头不利于视频结构的组织，所以有必要挖掘出视频的高层结构、如场景、幕等，从中得到视频镜头的结构语法和语义。

美国普渡大学的 Marzouk 等人采用视频结构挖掘的方法开发了视频内容结构和事件的挖掘框架，分别使用镜头分类、关键帧提取、镜头分组、组合并、场景聚类等方法把视频内容组织成 5 级层次结构。层的粒度大小按帧、镜头、组、场景、视频依次递增。视频结构的挖掘工作分三步进行：（1）组检测；（2）场景检测；（3）场景聚类。首先将视频镜头分割成语言丰富的单元，将空间上相邻、内容上相似的组归并为场景，扫描整个视频，过滤掉相似的场景。为实现镜头成组，应用特定的技术识别相似背景，在时间序列上识别有关的镜头，将时间或空间上相关的镜头分割为一个组，根据镜头特征，采用分类、聚类挖掘方法，将视频镜头组织成具有语言的单元－场景，它是由一系列相继的镜头组成，是在相同的地点拍摄的，具有相同的视频内容。

16.4.2　运动挖掘

运动是视频特有的特征，包括对象的运动和摄像机的运动。特别是对象运动信息尤为重要。视频技术和计算机视觉的快速发展，为运动特征的提取、分析、处理提供了有力的技术基础。利用计算机视觉方面的研究成果可以检测移动斑点、预测斑点运动轨迹、跟踪关键部位的运动等，从中提

取出对象运动的特征，并可进一步对运动模式进行包括移动对象轨迹索引、运动聚类、孤立点检测、关联分析等数据挖掘过程，并分析获得的时空数据特征，获得有价值的知识。视频运动挖掘的结果可以应用于交通调度、计算机辅助身体健康（理疗）、职业安全、人机工程等方面；有利于改善体育训练方法，医疗和诊断方法，而且，还可以提高计算机视觉和模式识别算法的性能。

16.4.3　趋势挖掘

运用统计归纳和关联等方法，挖掘视频中事件发生与时间发生关联的模式。根据事件持续的时间、事件发生间隔、事件序列片段等参数和特征，采用事件模式生长、频繁模式等方法进行数据挖掘，并通过分析视频运动特征及其随时间的变化情况，从而达到趋势分析和数据挖掘的目的。

16.5　音频挖掘

目前，数据挖掘对象很少涉及语音数据。这一方面是由于语音数据复杂，包含很多信息。例如音频中含基频信息、时长信息、幅度信息、位置信息以及重音信息等。同一个音节在不同的语句会表现出不同的信息特征，不同的语境会使音节自身的属性值发生变化。另一方面，语音数据挖掘的研究需要语音合成工作的技术积累。由于数据挖掘技术对处理对象的要求很高，因此不能处理直接录制音节的波形文件，必须对波形文件进行严格的预处理。例如，对录音波形进行音节和音节标注，这项工作需要大量的人力和物力资源，需要强大的语音处理能力和积累。将数据挖掘技术应用于语音信号处理可以解决部分现阶段较难解决的语音技术问题，同时尽可能减少人为经验因素对语音处理的影响，完成对语音处理从定性到定量的转变。因此，将数据挖掘方法应用于语音合成工作具有重要的意义和广阔的前景。

由于音频是像视频一样的连续媒体类型，音频信息的处理和挖掘与视频信息的检索和挖掘相似，很难有明确的界限。在音频数据挖掘中，可以使用语音转换和关键字抽取等技术音频数据转换为文本，然后挖掘文本数据。如图 16.7 所示。也可以使用音频信息处理技术筛选出关键语句并在筛选出的音频数据上直接挖掘。如图 16.8 所示。

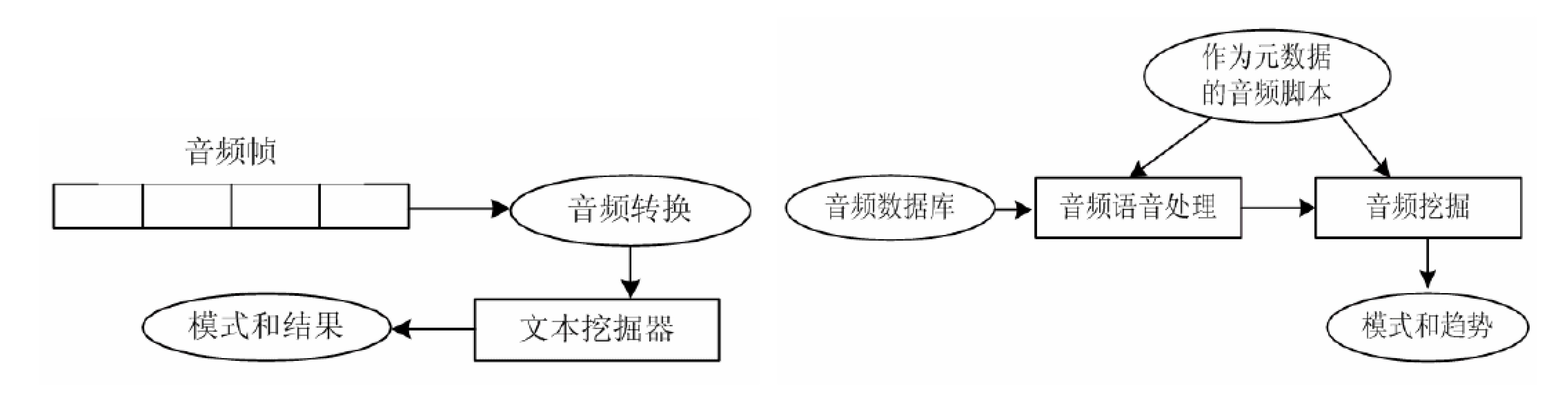


图 16.7　挖掘音频中的文本信息　　　　图 16.8　直接音频挖掘

总之，音频挖掘比视频挖掘难度更大。现有的音频挖掘系统基本上都是先把音频数据转换成文本数据，然后对文本数据进行挖掘。目前语音识别已广泛应用于 IVR（Interactive Voice Response，交互式语音应答），但 IVR 使用的语音识别技术相对简单。许多公司已经开发出音频识别软件，例如 IBM 的人类语音技术（HLT）等。目前的音频挖掘系统主要用于音频检索，比较著名的系统有美国 ScanSoft 公司的 AudioMining 开发系统，它将音频中的音频信息转换成文字信息，并对文字信息进行索引，由于文字信息和音频中原始音道的时间帧相关，通过用户提交的

文字信息索引就可以定位音频的播放位置。该软件包中的 AudioMining & XML Speech Indexing 可以根据音频和视频文件自动生成 XML 语音索引数据，实现网上音频搜索。

16.6　复合类型数据的挖掘

在实际应用过程中，多媒体数据挖掘的对象是由多种类型的媒体组成的复合体，例如文本和图像、文本和视频或文本、音频和视频等。如果对多媒体数据进行挖掘，则需要在两种或多种数据类型合成的基础上进行。

处理复合类型数据和处理异构数据库非常相似，异构环境中的数据库通常由多种类型的数据构成，可以采用两种数据挖掘方案。图 16.9 是先整合异构数据，然后在集成视图上挖掘；图 16.10 是先分别在各自的数据集上挖掘，然后再整合数据挖掘结果。无论采用哪种方案，多媒体分布式处理器都起着很重要的作用，如果采用先整合后挖掘的方案，整合须由 MDP 实现；如果采用先挖掘的方案，数据挖掘器要求扩展 MM-DBMS 的功能，挖掘结果则通过 MDP 整合。

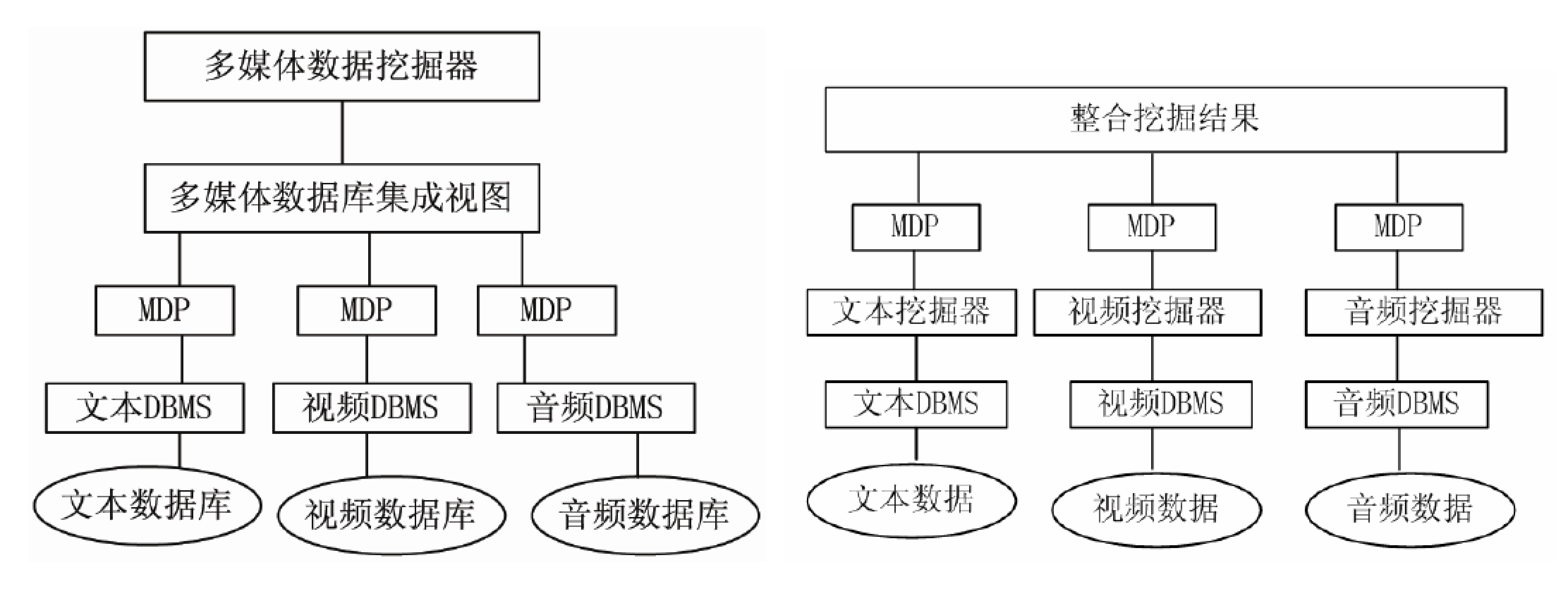


图 16.9　先整合后挖掘　　　　图 16.10　先挖掘后整合

由于单一数据类型的挖掘工作还有待于进一步研究，因此，针对多种数据类型复合数据的挖掘工作仍面临着很大挑战。

读书笔记

第 17 章

Web 数据挖掘技术

17.1 Web 数据挖掘技术概述

随着互联网技术的进一步发展，网上信息越来越多。目前网上的网页数量已达上百亿，而且正在以每月近千万的数量增长，甚至有人预言 Web 页面的数量每隔 100~120 天要翻一番。

Web 挖掘是挖掘与互联网有关的数据，既可以是网页包含的数据，也可以是 Web 操作产生的数据。数据可分为以下几类。

（1）网页本身的内容。

（2）网页内部结构，包括 HTML 或 XML 代码。

（3）网页之间的链接结构。

（4）描述网页被如何访问的使用数据。

（5）用户简档，包括与人口统计有关的信息，注册信息以及从 cookie 中获取的信息。

Web 挖掘任务可以分为多种，图 17.1 给出了一种挖掘活动的分类，内容可以包括文本或者图形数据。尽管不同的 Web 挖掘任务可以分开描述，但它们本质上是有联系的。

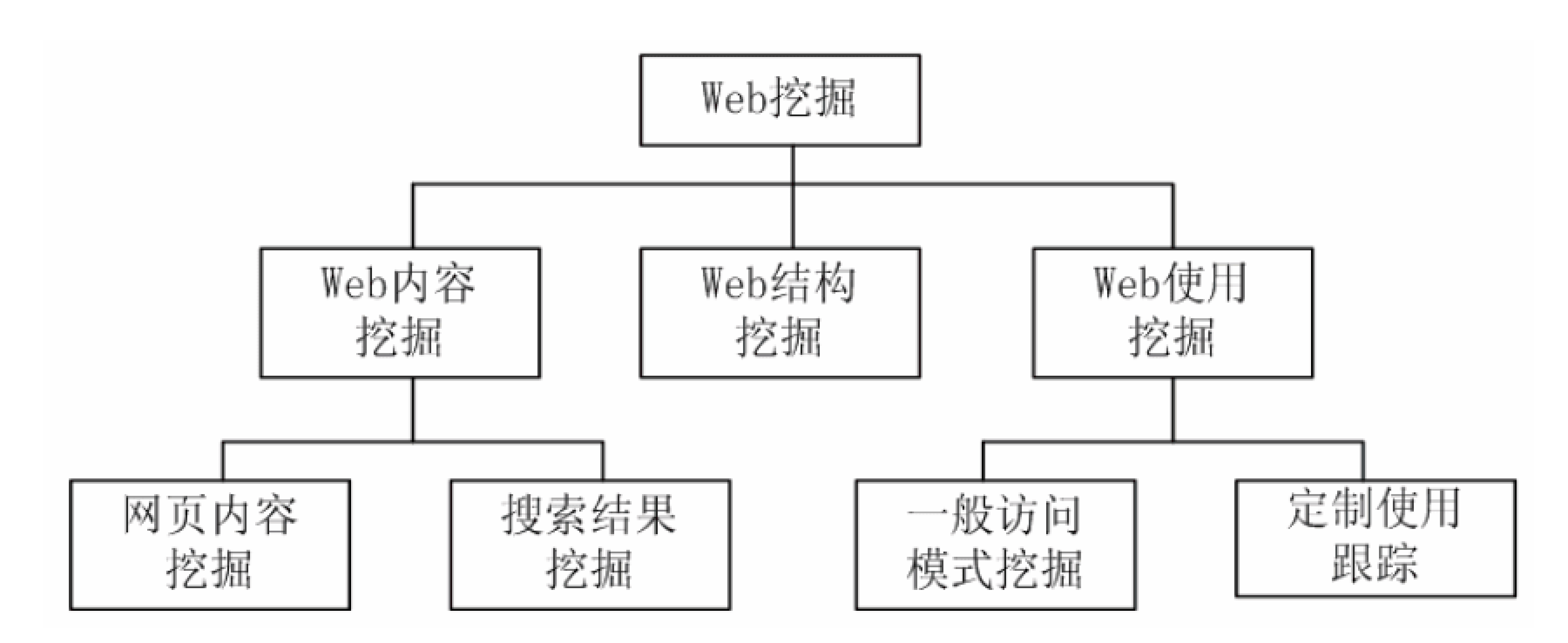


图 17.1　Web 挖掘

Web 挖掘有很多应用，其中一种应用是针对性广告技术（“瞄准”技术），即把广告发送给（而且只给）那些最有可能购买的潜在顾客。这样广告费用降低但不会影响效果。如果对特定地理区域的网民感兴趣，“瞄准”技术可以用来帮助把广告放在该区域的人们经常访问的网站上。通过分析网络访问记录，可以发现哪些站点对这个网站进行了访问，也能够得到访问者的信息，从而把广告出售给那些最受益的公司。

17.2 Web 内容挖掘

内容挖掘可以看作对基本搜索引擎所完成工作的扩展，数据挖掘技术可以用来帮助搜索引擎变得更迅速、更有效和具有更好的扩展性。有很多技术可以用来进行互联网搜索，多数是基于关键字的。使用概念层次、同义词、用户信息以及分析网页之间的链接可以使搜索引擎的效果得以改进。传统的搜索引擎使用爬虫（crawler）搜索互联网和搜集信息，用索引技术存储网页信息，使用查询处理为用户提供快速、准确的信息。

基本的内容挖掘是一种文本挖掘，文本挖掘的功能可以用一个层状结构表示，最简单的功能位于最上一层，最下层对应最复杂的功能。目前正在研究把自然语言处理技术用于文本挖掘，以

发现文本中隐含的语义，例问答系统。更传统的挖掘任务包括关键字搜索、相似性度量、聚类和分类等。

17.2.1　爬虫

机器人（蜘蛛、爬虫）是指遍历网页超文本结构的程序。遍历时初始的网页称为种子。从初始网页开始，所有指向外部的链接被保存在队列中，然后按顺序访问队列中的网页，这些网页包含的链接同样保存在队列中。机器人遍历网页时，就可以收集每个网页的信息，例如可以抽取关键字并保存在索引表中供使用该搜索引擎的用户使用。有的爬虫访问一定数量的网页后，会停下来建立索引，以替换旧索引。这种爬虫由于被周期性地激活，因此称为周期性爬虫。爬虫一般用来为搜索引擎建立索引，使索引在几乎没有人工干预的情况下基本上保持最新。

由于互联网规模巨大，产生了专用爬虫，它只访问与特定主题相关的网页。与传统的爬虫相比，使用许多专用爬虫能够覆盖更多的网页，并且随着 Web 规模的增长有更好的扩展性。专用爬虫结构包括三个主要组成部分：

- 该结构的主要部分是一个超文本分类器，对每个网页依据其与主题的相关程度打分。另外，该分类器对每个网页计算资源率，用于估计如果访问由该网页链接的其他网页所获得的收益大小。
- 提取器，用于确定中心网页。所谓中心网页是指包含若干相关链接的网页。中心网页很重要，它不一定包含与主题相关的信息，但是应该是顺着它的链接进行搜索。
- 爬虫，用于遍历互联网。访问网页的顺序依据一个优先级结构，优先级由分类器和提取器确定。

使用专用爬虫之前，用户先要准备一些感兴趣的示例网页。当用户浏览互联网时，就可以对感兴趣的网页进行标记。这些网页依据一棵有层次的分类树进行分类，树中节点如果与感兴趣的网页对应，则该节点标记为好。这些网页作为机器人搜索的起始网页，当搜索过程中发现相关网页时，需要确定是否访问它链向的其他网页，每个网页被分类为树中叶子节点。

后向爬虫用来发现那些本身与主题相关但未被已有的相关文档链接的网页。这些网页可能是新网页，或者是还没有被发现和被其他网页链接的网页。尽管网页中没有后向链接的信息，但后向爬行实现起来比较容易，因为多数搜索引擎都有此爬虫的信息。

17.2.2　虚拟 Web 视图

为了处理互联网上如此大量的无结构数据，一种途径是在网页数据（或者它的一部分）上建立多层数据库，这种数据库规模宏大并且是分布式的。数据库的每层比它的下层要更概括。多层数据库为部分互联网提供一种抽象的精简视图。可以构造多层数据库的一种视图，称为虚拟 Web 视图。

多层数据库建立索引时不使用爬虫，而是让网站服务人员（网站管理员、系统管理员）把索引（或者索引的变化）发送到执行索引任务的站点，这个过程在网站内容发生改变时触发。每层的索引比它下层的和它指向的层的索引要小。为了帮助建立多层数据库的第一层，需要使用抽取和翻译工具。翻译工具用来把网页文档转化为 XML 格式，抽取工具用来抽取想要的信息并插入多层数据库的第一层。

按层次往上，数据库更高层的分布性减弱，概括性变得更强。需要使用概化工具构造多层数据库的更高层，并且使用概念层次辅助概化过程。概念层次可以使用 WordNet 语义网络建立，它是一个英语语言数据库，包括名词、形容词、动词和副词，词划分为近义词组，词与词之间根据词法和语义关系连接起来。

WebML 作为一种数据挖掘语言被提出用于多层数据库上进行数据挖掘操作，它是 DMQL 的扩展。网页文档使用数据挖掘操作和关键词存取。

17.2.3 个性化

Web 内容挖掘的另外一个例子是个性化挖掘。使用个性化技术，网页访问或者网页的内容可以被更改从而更好地适应用户的需求。这涉及为每个用户创建独特的网页或者根据用户的要求决定搜索哪些网页。

使用个性化技术，发送给潜在客户的广告可以根据顾客的特定信息而有所选择。与"瞄准"技术不同，个性化挖掘可以在目标网页上进行，目的是引导顾客购买他（她）以前没找到的商品。个性化几乎和"瞄准"是相反的，"瞄准"是把广告放在别的站点上让用户能够访问到，而个性化是当用户访问站点时，针对这个用户设计广告。

个性化可以用许多方法实现，有些不是数据挖掘的方法。个性化挖掘需要研究日志数据从而可以发现用户访问行为的模式，这方面属于 Web 使用挖掘。

个性化挖掘可以看作一种聚类、分类甚至预测。使用分类，一个用户的要求可以根据一类用户的要求确定，也可以使用聚类根据一组与其他人相似的用户确定，最后使用预测来预知用户的真实要求。有三种基本类型的个性化技术：

- 使用手工技术进行个性化，例如通过用户注册时的偏好选择，基于用户档案或统计信息建立的规则对用户分类。
- 协同过滤技术，把以前相似用户评价较高的信息（网页）推荐给用户。
- 基于内容的过滤，指基于网页内容以及用户简档信息之间的相似性搜索网页。

My Yahoo!是最早使用个性化技术的网站之一，使用 My Yahoo!站点时，用户可以自己对屏幕显示的内容进行个性化配置，可以选择配置天气、新闻、股市报价、电影和体育等信息。配置偏好信息后，每当用户登录时，其配置的页面就会显示。个性化工作由用户完成，明确反映用户想看到的信息。

可以使用兴趣度确定用户是否对负面感兴趣。兴趣度是基于网页内容和用户需求的相似度。通过在网页和专为用户创建的配置中共同出现的词来度量相似度。确定兴趣度可按两级方案进行。第一级基于用户最近读过的文章，第二级是关于用户长期的、一般兴趣的描述。一个网页如果与二者之一足够接近就被认为是有趣的。

由于人们在决策时通常会参考其他人的意见，如果一个人喜欢某个电视节目，这个人的朋友也可能喜欢。因此可以根据相似用户的喜好预测某个用户的偏好，这可以看作是一种聚类。这种技术用在 Web 挖掘中被称为协同过滤。

17.3　Web 结构挖掘

Web 结构挖掘可以看作是为互联网的组织建立一个模型，用来对网页分类或者为网页建立相似性度量。

17.3.1　PageRank

PageRank 算法用于提高搜索引擎的搜索效果和效率。度量网页的重要性以及为传统搜索引擎使用关键字搜索的结果进行优先级排序。网页的 PageRank 值通过指向它的网页计算，这实际上是基于网页后向链接的一种度量。给定网页 p，用 B_p 表示指向 p 的网页集合，F_p 表示由 p 指向其他网页的链接集合。网页 p 的 PageRank 值定义为

$$PR(p) = c\sum_{q\in B_p}\frac{PR(q)}{N_q}$$

这里 $N_q = |F_q|$，常数 $c\in[0,1]$，用于归一化。

当网页之间的链接产生环路时（网页 A 指向 B，同时 B 指向 A），PageRank 值的计算会出现所谓的排序沉没问题，此时这些网页的 PageRank 值会增加。可用下式解决

$$PR'(p) = c\sum_{q\in B_p}\frac{PR(q)}{N_q} + cE(v)$$

这里 c 取最大。$E(v)$项对应一个虚拟链接，用于模拟一个随机的用户周期性地决定访问链接的网页而跳到一个新的网页。$E(v)$为在每对节点之间添加小概率的链接。

17.3.2　Clever

IBM 公司开发的 Clever 系统，其目标是发现权威网页和中心网页。权威网页是指对请求的信息来说是“最好的源”的网页，含有指向权威网页链接的是中心网页。系统使用加权技术识别权威网页和中心网页。

由于网站的分散和无监督开发等特点，用户无法知道网页包含的信息是否正确。目前，没有办法防止用户制作包含错误或者含有谎言的网页。另外，有些网页的质量可能高于其他网页，这些网页通常被看作是最权威的。

HITS 算法可以用来搜索中心网页和权威网页，该技术包含两部分：

- 基于一组给定的关键词（从查询中得到），发现一组（可能数以千计）相关网页。
- 针对这些网页进行权威性度量和作为中心网页的度量，返回度量值最高的网页。

17.4　Web 使用挖掘

Web 使用挖掘的研究对象是 Web 使用数据或者 Web 日志。Web 日志是一列网页访问数据。由于每一项数据对应鼠标的一次点击，Web 日志有时候称为点击流数据。可以从客户或者服务器的观点对日志进行分析。当从服务器的观点分析时，挖掘发现的是提供服务的网站的信息，挖掘的结果可以帮助改善网站的设计。通过分析客户的点击序列，可以发现一个（或者一组）用户的信息，这些信息可以帮助实现网页的预存取和缓存。

Web 使用挖掘可以应用于多种不同目的：

- 通过跟踪用户以前访问的网页，实现用户的个性化。这些网页可以用来识别用户的典型搜索行为，以及用于预测用户以后想访问的网页。
- 通过确定用户的频繁访问行为，可以访问用户需求的链接，提高用户将来访问的总体性能。
- 关于哪些网页经常被访问的信息可以用于缓存。
- 识别通常的访问行为不但可以帮助改变网站的链接结构，还可以帮助改进网页的设计和对网站进行其他修改。
- 使用模式可以用来收集商业智能信息，以提高销售和改进广告效果。
- 搜集用户如何访问网页的统计信息可以看作是挖掘的一部分，也可以看作不是。

Web 使用挖掘包含三种类型的工作：

- 预处理工作，集中对日志数据的格式进行转化。
- 模式发现工作是整个挖掘过程的主要部分，它从日志数据中发现隐含的模式。
- 模式分析是研究和解释模式发现工作的结果。

17.4.1　预处理

Web 使用日志的格式可能不适于挖掘程序，所以需要对其进行预处理，包括清洗、用户识别、会话识别、路径补全和转换格式。

清洗是清洗日志中无关的信息，如包含图形（gif、jpg 等）的日志项可以删除。因为代理服务器、客户端缓存和企业防火墙的广泛使用，使得用户识别是一个较为复杂的问题。尽管对网页的一次访问包括源 URL 或 IP 地址以说明请求的来源，但这却不能保证完全正确地确定用户的位置。通过网络服务商上网的用户对应的源位置都是该服务器的，对单个用户来说位置不唯一。另外，同一用户会使用不同的网络服务商。同样，在同一段时间内，会有来自同一台机器的若干用户访问网页。不管用户使用哪台计算机，用 cookie 可以帮助识别访问网页的用户。

如果客户端使用缓冲，就难以识别用户访问的网页序列。在这种情况下，服务器端的日志会丢失用户访问的网页。可以通过预测丢失的网页补全日志。路径补全技术可以把实际发生但日志中没有的访问记录添加到日志文件中。

17.4.2　数据结构

在 Web 使用挖掘过程中几种数据结构用来记录识别出的模式。其中一种基本的数据结构称为 trie，它是一种树结构，树中从根节点到叶子节点的每条路径表示一个序列。trie 在模式匹配应用中用于存储字符串，字符串的每个字符保存在节点的边中，字符串的前缀是共享的。

17.4.3　模式发现

点击流数据的挖掘一般是发现浏览模式。浏览模式是在一个会话中用户访问的一组网页。Web 使用挖掘也能够采用其他类型的模式。相似的浏览模式可以汇总在一起用于用户聚类，这跟网页聚类不同，网页聚类是识别相似的网页而不是用户。

浏览模式主要有关联规则、片段、顺序模式、极大前向访问、极大频繁序列等，它们之间的

区别可以用下面的特征来描述：

- 是否允许重复的网页访问（后向访问和刷新）。
- 模式可以只包含连续的网页访问，或者同一会话的任何网页。
- 访问模式在会话中是否极大。频繁模式极大是指模式中不包含任何频繁子模式。

上述三种特征的不同组合可以用来发现不同的模式，因此可以用于不同目的。有关连续网页频繁访问的知识可以用来预测后来的网页访问，从而可以用于网页预取和缓存。关于后向访问的知识可以用于改进一组网页的设计，通过添加新的链接缩短访问时间。模式的极大性主要用于减少模式的数量。

17.4.4　模式发现

模式一旦被发现，通常需要进行分析以确定其如何使用。用户可能对有些模式不感兴趣，需要将它们删除。

通过对网站日志的分析，不仅可以发现频繁出现的模式，还可以识别出用户对哪些模式感兴趣。发现的模式也可以不是邻近网页的。

将网站的访问者分为短期访问者、调查者和顾客，以比较电子商务网站的顾客和非顾客用户的浏览模式。首先通过预处理过滤掉短期访问者，然后使用概念层次把网页内容抽象到更一般的概念，把日志划分为顾客和非顾客的，最后按照特定的要求分析每部分日志以寻找模式，最后对每部分找到的模式作相似性比较。相似性按照下面的规则确定：

如果两个模式的 g-序列最开始至少 *n* 个网页相同，那么这两个模式是相似的。这里 *n* 由用户指定。g-序列是一个向量，向量中的元素不仅包括访问的网页，还可以包含通配符。例 g-序列 *b**c* 代表 *b* 网页开头，*c* 网页结束，中间可以包含任意数目的任何网页。由于使用通配符，因此网页可以不邻近。更加复杂的 g-序列可以对通配符代表的网页数目进行限制。

另外，可以只考虑频繁出现的模式片段，这样做的目的是增加顾客的数目。如果发现非顾客的模式，并且没有相似的顾客模式，则表明网站的链接结构或者网页的内容设计需要改变。

17.4.5　基于组织协同进化的 Web 日志挖掘算法

Web 日志挖掘是对一个或若干个网站的用户访问记录和其他数据组成的数据集进行分析挖掘，并从中获得有价值的有关网站访问情况的模式知识。

目前，Web 日志挖掘技术主要有以 Han 为代表的基于数据立方体的方法和以 Chen 为代表的基于 Web 事务的方法。在基于数据立方体的 Web 日志挖掘技术中，Han 等人根据 Web 服务器日志文件建立数据立方体，然后对数据立方体进行数据挖掘和联机分析处理。在基于 Web 事务的 Web 日志挖掘中，Chen 等人将数据挖掘技术应用于服务器日志中。而基于组织协同进化的 Web 日志挖掘算法可以克服传统算法中的不足。

典型的服务器日志包括以下信息：IP 地址、请求时间、方法（如 get），被请求文件的 UR1，HTTP 版本号、返回码、传输字节数，引用页的 UR1 和代表。对 Web 日志进行预处理后，可以 log={ip,uid,url,time}的形式表示 Web 服务器日志。其中，ip、uid、url、time 分别代表客户 IP、客户 ID、客户请求的 UR1 和浏览时间。然后对日志数据再作进一步的处理，使其能合理地反映用户在某一段时间内的浏览行为。

一个网页一般含有成百上千的页面，因此仅仅对页面进行分析很难发现有用的信息。但网页的页面一般是按页面的类别进行组织的，比如一个新闻网站会将页面按国际新闻、国内新闻、经济、政治等栏目分类组织，所以为了便于对 Web 日志进行数据挖掘，可以用表的形式来表达日志数据集，表的结构如表 17.1 如示。

表 17.1　Web 日志数据的表示

url•type₁	url•type₂	…	url•typeₙ	user•type
1	0	···	1	a
1	1	···	1	b
1	1	···	0	c
0	1	···	1	d

表中的每一行记录表示用户的一次会话，其中 $url•type_i(i=1,2,\ldots,n)$表示第 i 个网页类型，user•type 表示用户的类型。其中的 $url_i•type$ 表示网页类别，type 表示客户的类型。当每次用户会话中，有属于 $type_i$ 的网页被访问，该字段的值就为 1，否则为 0。

算法中处理对象是组织，它由一条或多条日志记录组成，可分为自由态组织、异常态组织和正常态组织 3 种。自由态组织是指包含日志记录个数为 1 的组织，其属性均为有用属性，其集合记为 free。异常态组织是指有用属性集为空的组织，其集合记为 abnormal；其余的组织为正常态组织，其集合记为 normal。在组织中所有用户访问记录的取值均相同的条件属性为相同属性；如果某条件属性为相同属性，且按一定规则该条件属性被判为可参与组织适应度的计算，则该条件属性为有用属性。

算法中，各个条件属性的重要度随着种群的不断进化也不断进化。属性的重要度在进化的过程中，根据不同的情况而降低和升高。算法中的算子为增减算子、交换算子、合并算子和组织选择算子，但算子中的 m 与 n 定为一百分数。这样当组织大小变化时，组织中参与增减算子与交换算子操作的对象个数也随之变化。另外，当随机选择的两个组织，假如其中有一个组织中日志记录个数不大于 1，则只执行合并算子。

在种群进化的过程中，先通过两个不同的组织随机地执行增减、交换或合并算子产生子代组织；然后用组织选择算子从父代和子代中选择出组织适应度高的组织，并使整个种群的适应度不断提高。整个种群就通过这样的方式不断进化，当进化结束后，从最终进化的组织中提取规则。算法中正常态组织的适应度计算公式如下

$$\text{fitness}_{\text{org}} = |\text{org}| \prod_{i=1}^{|\text{use}_{\text{org}}|} ci_i$$

其中：ci_i 表示组织 org 有用属集中第 i 个属性的重要度。在组织选择中，用下面的公式计算父代和子代组织的适应度值

$$\text{fitness} = \max\{\text{fitness}_{\text{org1}}, \text{fitness}_{\text{org2}}\}$$

其中：$\text{fitness}_{\text{org1}}$、$\text{fitness}_{\text{org2}}$ 分别表示组织 org1 和组织 org2 的组织适应度值。

算法的具体描述如下。

① 群体初始化：把网站用户的类型定义为 $d_t(t=1,2,\cdots,m)$。这样，通过把每一类用户类型定义为一个种群，得到 m 个种群。其中种群定义为 $p_i(i=1,2,\cdots,m)$。把用户类型为 d_i 的日志记录以自由态组织加入种群 p_i 中，且令进化代数 $t=0$，变量 $i=1$。

② 如果变量 i 大于种群个数 m，则转入步骤⑧，否则转入步骤③。

③ 如果在当前进化代数 t 中，种群 $p_i(t)$中未进化的组织个数大于 1，则转入步骤④，否则转步骤⑦。

④ 从 $p_i(t)$中随机选择两个组织 org_{p1} 和 org_{p2} 作为父代组织；当父代组织中有一组织为自由态组织时，执行合并算子；否则从增减、交换和合并算子中随机选择一个算子，对 org_{p1} 和 org_{p2} 进行相应的操作，产生子代组织 org_{c1} 和 org_{c2}。

⑤ 组织适应度的计算：若组织所含的日志记录个数为 1，则令组织类型为 free，组织适应度为 0；否则根据属性重要度的进化算法，更新属性重要度，并确定有用属性集合；若有用属性集合为空集，则令组织类型为 abnormal，组织适应度为 − 1；否则令该组织类型为 normal，并计算适应度。

⑥ 组织选择：计算父代和子代组织的适应度值。若父代组织的适应度大于子代组织的适应度，将 org_{c1} 和 org_{c2} 淘汰，将 org_{p1}、org_{p2} 标志记为已进化，然后加入下一代；否则，将 org_{p1} 和 org_{p2} 淘汰，如果 org_{c1} 、$org_{c2} \notin abnormal$ ，将 org_{c1} 和 org_{c2} 标志为已进化，然后加入下一代；否则，不妨设异常态组织为 org_{c2}，并将 org_{c2} 解散，其对象以自由态组织形式进入下一代，将 org_{c1} 标志为已进化，然后加入下一代。

⑦ 变量 i 的值加 1（即对下一个种群执行组织进化操作），转入步骤②。

⑧ 如果进化代数 t 达到了设定的进化代数，则用相应的规则提取算法从最终进化的组织中提取规则，返回；否则，对变量 i 赋初值 1，进化代数的值加 1，转入步骤②。

当进化结束后，每一个种群中具有相同抽取规则的日志记录就聚集在一个组织中。通常可以简单地将每个组织的相同属性转化成规则，这样从每个组织中可以得到一条规则；但当一个种群进化结束后形成多个组织时，这样简单地提取规则形成的规则集合会有较大的冗余。这时如果某个组织的有用属性集为另一组织有用属性集的子集，则将这两个组织合并，新组织的有用属性集为原来两个有用属性集的交集。

该算法的计算复杂度低，具有较快的收敛速度，算法产生的规则集较小，而且预测的正确率较高。

读书笔记

第 4 篇　数据挖掘应用实战

数据挖掘从一开始就是面向应用的，自从20世纪80年代数据挖掘出现至今，数据挖掘在理论研究上日臻成熟，正不断扩展其应用范围，该技术已经在电信、金融、医疗保健、商业、入侵检测、工程与科学等很多领域中得到了广泛的应用，出现了大量的商品化的数据挖掘产品和系统。

目前，国内已经存在的数据挖掘相关产品大致属于以下三种类型。

（1）面向某一行业甚至某一应用的专用数据挖掘产品。这类产品是由开发商为某一特定用户或特定应用开发的专用数据挖掘系统衍化而来。其特点是算法针对性强、模型设计严谨科学、功能较强；缺点是通用性较差，不易修改。

（2）基于国外产品经过二次开发而来的软件。国内一些数据挖掘产品是在SAS、SAP等国外产品基础上经过二次开发得到的。从严格意义上讲，这些软件不具有完全的自主知识产权。

（3）自主研发的通用数据挖掘产品。国内也有少量的自主开发的数据挖掘软件。这些产品中包含的算法多，可用范围广，可修改，维护性较强，但规范性较差，未遵守国外数据挖掘业界的工业标准，造成扩展性及兼容性的缺陷。

随着数据挖掘的需求越来越强，如何缩短国内数据挖掘产品与国外产品在数量上和质量上的差别，研发相应的数据挖掘软件成为国内业界的一个重要问题。

一个性能较为完善的数据挖掘软件应具有较好的可扩展性、可重用性、易用性等性能。根据这个目标，可以采用多种语言进行编程。

MATLAB 软件是一种功能强、效率高、便于进行科学和工程计算的交互式软件包，编程效率高，易学易用，也易于与其他传统编程语言（如C、C＋＋和Fortran）互为调用。自推出后，即风行美国、流传全世界，被广泛应用于信号和图像处理、通信、控制系统设计、测试和测量、财务建模和分析以及计算生物等众多领域的数学与计算、算法开发、数据采集、建模与模拟、数据分析、研究和可视化、科学与工程图形、应用程序开发等实际应用。

正是由于MATLAB具有如此强大的功能，本篇将介绍基于MATLAB的数据挖掘在科学研究中的应用，为推广数据挖掘的应用以及开发数据挖掘软件提供一个有力的工具和重要的参考作用。

第 18 章

数据统计特性

18.1 数据关系发现

数据挖掘基于海量数据，而这些海量数据大多都存储在大型的关系数据库中。所以在进行数据挖掘之前，首要的是分析数据库中的数据业务关系，数据技术关系和数据描述等。

数据库业务关系指数据库表间及表内关系。数据挖掘所基于的数据可能是存在于同一数据文件、不同数据文件或不同的数据库中，因此，表间关系就是指各个数据库中数据相互间的关系，如两个数据库表间的数量是依赖关系还是从属关系等。而表内关系是要发现数据库表内字段间的关系，主要探索列属性、列逻辑相关等。在此基础上提取元数据，包括字段名称、字段数据类型等一系列相关的内容，从而为数据挖掘准备好数据。

数据统计又称为汇总统计，即用单个数或数的小集合来捕获大的数据集的各种属性或特征。数据统计特性主要有中心趋势和离散程度两部分。中心趋势度量包括均值、中位数、众数和中列数；离散程度度量有四分位数、四分位数极差、频率、方差等。

18.2 频率和众数

设一个在$\{x_1,x_2,\cdots,x_k\}$上取值的分类属性 x 和 m 个对象的取值，值 x_i 的频率定义为

$$\text{frequency}(x_i)=\frac{\text{具有属性值}x_i\text{的对象数}}{m}$$

众数是集中出现频率最高的值。对于分类属性而言，众数可以看成中心趋势度量；对于连续属性而言，众数则通常没有意义。

18.3 百分位数（percentile）

对于有序数据，有时考虑值集的百分位数更有意义。给定一个有序的或连续的属性 x 和 0 到 100 之间的数 p，数据集合的第 p 个百分位数 x_p 是一个 x 值，使得 x 的 p%的观察值正好小于 x_p。中位数是第 50 个百分位数 $x_{50\%}$。

除中位数外，最常用的百分位数是四分位数（quartile），第一个四分位数记作 Q_1，是第 25 个百分位数 $x_{25\%}$；第三个四分位数记作 Q_3，是第 75 个百分位数 $x_{75\%}$。四分位数可以给出数据分布的中心、离散和形状的某种指示，第一个和第三个四分位数之间的距离是分布的一种简单度量，它给出被数据的中间一半所覆盖的范围，称为四分位极差，定义为 $\text{IQR}=Q_3-Q_1$。

18.4 中心度量

数据集“中心”的最常用、最有效的数值度量是均值和中位数。

设属性 x 的 m 个值为$\{x_1,x_2,\cdots,x_m\}$，$\{x_{(1)},x_{(2)},\cdots,x_{(m)}\}$代表以非递减排序后的 x 值，该属性值的均值和中位数分别定义为

$$\text{mean}(x_i)=\bar{x}=\frac{\sum_{i=1}^{m}x_i}{m}=\frac{x_1+x_2+\cdots+x_m}{m}$$

$$\text{median}(x_i)=\begin{cases}x_{(r+1)} & \text{如果是奇数，即 } m=2r+1\\ \dfrac{1}{2}(x_{(r)}+x_{(r+1)}) & \text{如果 } m \text{ 是偶数，即 } m=2r\end{cases}$$

有时，集合中每个值 x_i 与一个权值 w_i 相关联，权值反映对应值的显著性、重要性或出现频率，此时常使用加权算术平均值

$$\bar{x}=\frac{\sum_{i=1}^{m}w_i x_i}{\sum_{i=1}^{m}w_i}=\frac{x_1w_1+x_2w_2+\cdots+x_mw_m}{w_1+w_2+\cdots+w_m}$$

尽管均值是描述数据集中最常用的单个度量，但不一定是度量数据中心的最合适的参数。均值的主要问题是对极端值（如离群值）很敏感，即使少量极端值也可以影响均值。为了减少极端值的影响，可使用截断均值。它是指定 0 到 100 之间的百分位数 p，丢弃高端和低端$(p/2)\%$的数据，然后用常规的方法计算所得的均值结果即为截断误差。

对于倾斜的（非对称的）数据，数据中心的一个较好度量是中位数。在完全对称的数据分布中，均值和中位数具有相同的值；如果是正倾斜的，其均值大于中位数；如果是负倾斜的，则均值小于中位数。

中列数（midrange）也可以用来评估数据的中心趋势，其定义为

$$\text{midrange}(x)=\frac{1}{2}(\max(x)+\min(x))$$

18.5　散布程度度量

连续数据的另一个常用汇总统计量是值集的散布度量，它表示属性值是否散布很宽，或者是否相对集中在某个值附近。

最简单的散布度量是极差，其定义为

$$\text{Range}(x)=\max(x)-\min(x)$$

尽管极差可以标识最大散布，但是如果数据的大部分值都集中在一个较窄的范围内，极端值的个数相对较少，则可能会引起误解。此时采用方差较为合适

$$\text{variance}(x)=S_x^2=\frac{1}{m-1}\sum_{i=1}^{m}(x_i-\bar{x})^2$$

方差对离群值很敏感，更加稳健的值集散布估计方法有绝对平均偏差（Absolute Average Deviation，ADD）、中位数绝对偏差（Median Absolute Deviation，MAD）和四分位数极差 IRQ。

$$\text{ADD}=\frac{1}{m}\sum_{i=1}^{m}|x_i-\bar{x}|$$

$$\text{MAD}=median(\{|x_1-\bar{x}|,\cdots,|x_m-\bar{x}|\})$$

还可以用偏度和峰度来刻画数据的偏态、尾重程度的度量。偏度的计算公式为

$$G_1=\frac{1}{(m-1)(m-2)s^3}\sum_{i=1}^{m}(x_i-\bar{x})^3=\frac{m^2u_3}{(m-1)(m-2)s^3}$$

式中：s 是标准差，$u_k=\frac{1}{m}\sum_{i=1}^{m}(x_i-\bar{x})^k$ 是样本 k 阶中心距。

偏度是刻画数据对称性的指标。关于均值对称的数据其偏度 $G_1=0$；右侧更分散的数据（即右尾长）偏度为正（$G_1>0$）；左侧更分散的数据（左尾长）偏度为负（$G_1<0$）。

峰度的计算公式为

$$\begin{aligned}G_2&=\frac{m(m+1)}{(m-1)(m-2)(m-3)s^4}\sum_{i=1}^{m}(x_i-\bar{x})^4-3\frac{(m-1)^2}{(m-2)(m-3)}\\&=\frac{m^2(m+1)u_4}{(m-1)(m-2)(m-3)s^4}-3\frac{(m-1)^2}{(m-2)(m-3)}\end{aligned}$$

当数据的总体分布为正态分布时，峰度 G_2 近似为 0；当分布较正态分布的尾部更分散时，峰度为正（$G_2>0$）；否则峰度为负（$G_2<0$）。当峰度为正时，两侧极端数据较多（粗尾）；当峰度为负时，两侧极端数据较少（细尾）。

18.6 数据的分布描述

数据的数字特征刻画了数据的主要特征，而要对数据的总体情况作全面的描述，就要研究数据的分布。对数据分布的主要描述方法是频数或频率分布表、直方图、总体分布、茎叶图等。

1. 频数频率分布表

（1）当数据为离散型时。

设样本观察值$\{x_1,x_2,\cdots,x_n\}$，$\{x_{(1)},x_{(2)},\cdots,x_{(l)}\}$（$n=\sum_{i=1}^{l}m_i$）代表以非递减排序后的 x 值，则可以得到表 18.1 所示的频数分布和频率分布表。

表 18.1　频数分布和频率分布表

x	$x_{(1)}$	$x_{(2)}$	…	$x_{(l)}$
频数 m_i	m_1	m_2	…	m_l
频率 m_i/n	m_1/n	m_2/n	…	m_2/n

（2）当数据为连续型时。

数据取值为一有限区间[a,b)，通常将[a,b)分成 l 个（l<n）区间（一般是等间距的），每个区间的长度为 $(b-a)/l$（称为组距），则

$$a=a_0<a_1<a_2<\cdots<a_{l-1}<a_l=b$$

通常组数可以考虑取：$l\approx1.87(n-1)^{\frac{2}{5}}$

表 18.2 给出了一些 l 值以供参考。

表 18.2　数据分组数的参考值

n	40~60	100	150	200	400	600	800	1000	1500	2000	5000	10 000
l	6~8	7~9	10~15	16	20	24	27	30	35	39	56	74

组距 Δx=（数据中最大值－数据中最小值）/组数，各组区间的端点为

$$a_0, a_0+\Delta x=a_1, a_0+2\Delta x=a_2,\cdots,a_0+l\Delta x=a_l$$

区间为

$$[a_0,a_1),[a_1,a_2),\cdots,[a_{l-1},a_l)$$

其中：a_0 可略小于数据的最小值；a_1 可略大于数据的最大值。

通常可用每组的组中值 =（组上限 + 组下限）/2 来代表该组的变量取值。

统计样本数据落入每个区间的个数－频数，并列出其频数、频率分布表。

2. 直方图

对于数据分布，常用直方图进行描述，即频率分布的图形表示。它在组距相等场合，常用宽度相等的长条矩形表示，矩形的高低表示频率的大小，横坐标为变量的取值范围，纵坐标为频数。若把纵坐标改为频率就得到频率直方图。

为使诸长条矩形面积和为 1，通常将纵坐标取为频率/组距（$f_i/\Delta x$），如此得到的直方图称为单位频率直方图或简称频率直方图。

3. 经验分布函数

直方图的制作较适合于连续型分布的场合。对于一般总体分布，若要估计它的总体分布函数 $F(x)$，可以用经验分布函数进行。设样本观察值 $\{x_1,x_2,\cdots,x_n\}$，$\{x_{(1)},x_{(2)},\cdots,x_{(l)}\}$（$n=\sum_{i=1}^{l} m_i$）代表以非递减排序后的 x 值，则经验分布函数为

$$F_n(x)=\begin{cases}0, & x<x_{(1)}\\ \dfrac{k}{n}, & x_{(i)}\leqslant x<x_{(i+1)}\\ 1, & x\geqslant x_{(n)}\end{cases}$$

4. QQ 图

不论是直方图还是经验分布图，要从图上鉴别样本是否近似于某种类型的分布是困难的。QQ 图则可以鉴别样本的分布是否近似于某种类型的分布。

现假设总体分布为正态分布 $N(\mu,\sigma^2)$，对于样本 $\{x_1,x_2,\cdots,x_n\}$，$\{x_{(1)},x_{(2)},\cdots,x_{(l)}\}$（$n=\sum_{i=1}^{l} m_i$）为以非递减排序后的 x 值，设 $\Phi(x)$ 是标准正态分布 $N(0,1)$ 的分布函数，$\Phi^{-1}(x)$ 是其反函数，对应正态分布的 QQ 图是由以下的点构成的散点图：

$$\left(\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right), x_{(i)}\right),\ 1 \leqslant i \leqslant n$$

若样本数据近似于正态分布，则 QQ 图上的点近似在直线 $y=\sigma x+\mu$ 附近。所以可以利用QQ图检验样本数据是否来自正态分布总体。利用QQ图还可以获得样本偏度和峰度的有关信息。

5. 茎叶图

把第一个数值分成两部分，小数点前面（或大整数）部分称为茎，小数点后面（或小整数）部分称为叶，然后用一条竖线，在竖线的左侧写上茎，右侧写上叶，就形成了如图 18.1 所示的茎叶图。

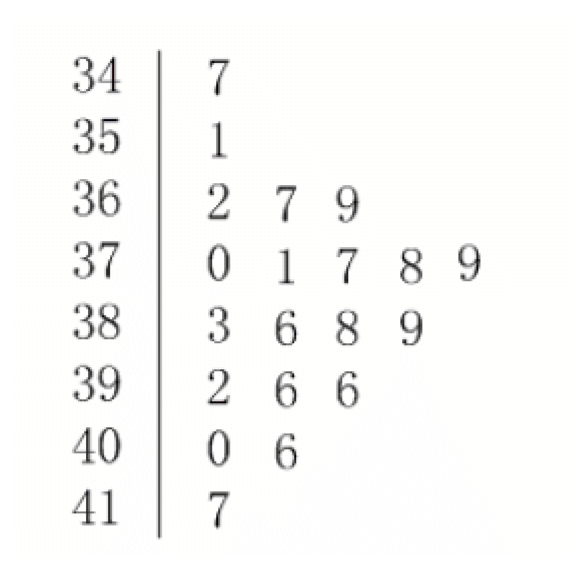


图 18.1　茎叶图

6. 盒形图

盒形图能直观、简洁地展现数据分布的主要特征，其构造方法如下。

（1）画一个箱子，其两侧恰为下四分位数 Q_1 和上四分位数 Q_3，中间有一道线，是中位数 M 的位置。这个箱子包含了样本中 50%的数据。

（2）在箱子上下两侧各引出一条水平线，分别延伸至异常值截断点，异常值用“+”表示。

图 18.2 即为一盒形图。

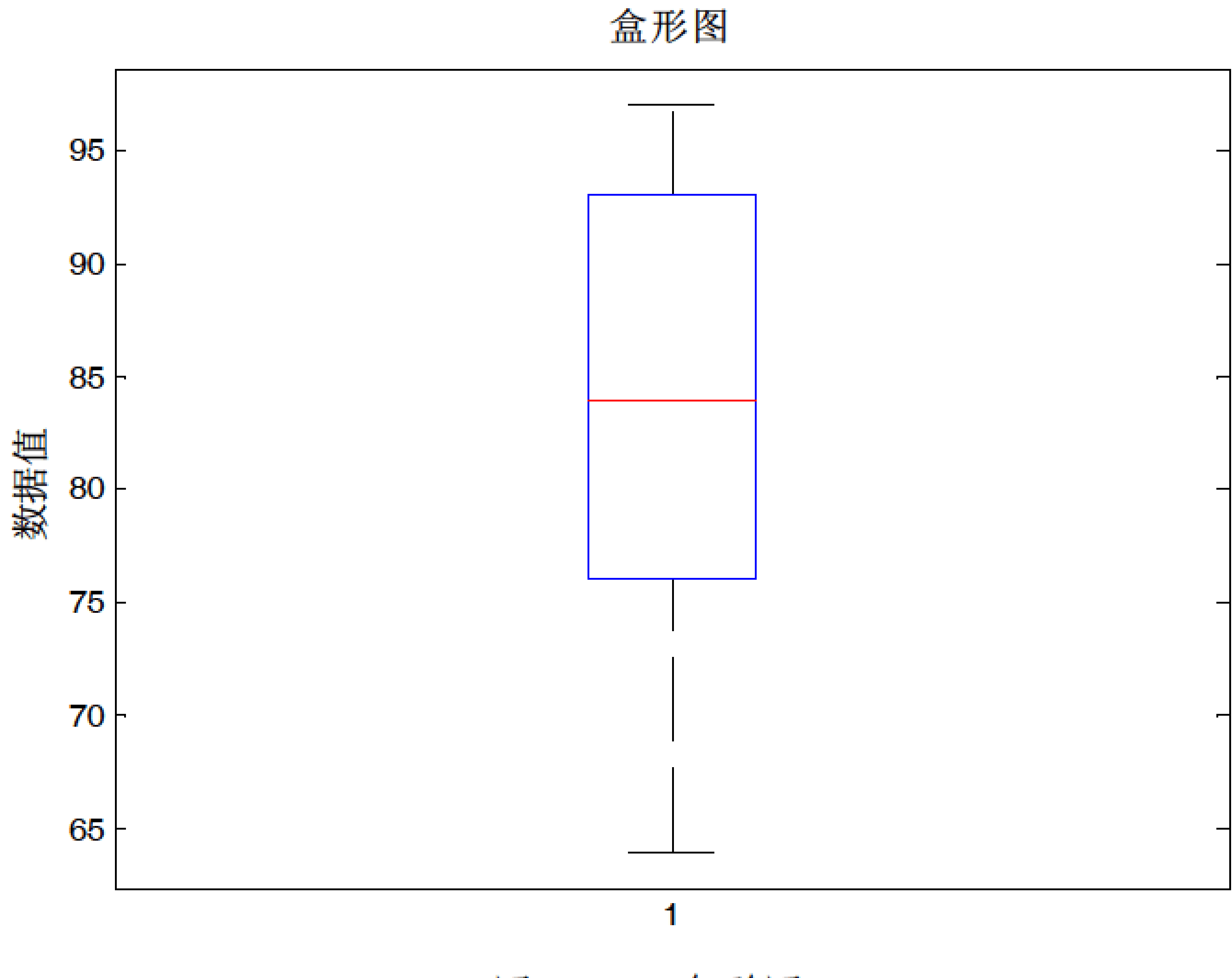


图 18.2　盒形图

18.7　数据的概率分布

数据概率分布主要是指数据的自然属性。不同的维度、变量下的数据分布可能不同，不同种类的数据有不同的分布。

常见的连续数据分布有高斯分布、T 分布、F 分布、二次项分布、几何分布和泊松分布等。这些分布的具体描述见统计学相关书籍。

读书笔记

第 19 章

数据预处理

19.1 数据预处理完毕

数据挖掘的目的是从大量日常业务数据中抽取一些有价值的知识或信息。原始业务数据是知识和信息提取的源泉，对于数据挖掘十分重要。在实际应用中，海量数据来自众多系统，具有多种形式和类型，其总是杂乱无章、不完全的，有时是难以理解的。数据的质量主要受噪声数据、空缺数据、冗余数据、模糊数据、无意义数据以及不一致数据等方面的影响，另外，原有数据特征有时不能足够地体现隐藏在其后的规律，需要从原有特征探索新的特征，以更好地表现对象的行为规律。因此，在数据挖掘前有必要通过预处理来提高数据的“质量”。

数据预处理技术为进一步的数据分析做准备，并能确定挖掘的类型，可以提高数据挖掘的质量。其中，数据清理可以纠正不一致数据，去掉数据中的噪声；数据集成能将多个数据源合并成一致的数据存储模式，如数据立方体；数据转换可以把数据变换成适于数据挖掘的形式；数据归约通过聚集、删除冗余特性或聚类等方法来压缩数据。

19.2 数据清理

数据清理的目的是检测数据中的错误和不一致，通过填写缺失的数据、光滑噪声数据、识别或删除离群点并解决不一致的现象，从而改善数据质量，提高数据挖掘的精度和性能。

19.2.1 填补缺失数据

缺失值是指本该有但却没有的数据。一个对象遗漏一个或多个属性值并不少见，缺失值并不意味着数据有错误，其产生的原因多种多样。

（1）有些信息暂时无法获取。例如在医疗数据库中，并非所有病人的所有临床都能在给定的时间内得到，这样就使一部分属性值空缺出来。有些信息（如年龄、体重、收入等）则是由于涉及个人隐私而无法得到。

（2）某些属性并不能用于所有对象，即对于某个对象来说，该属性值是不存在的。例如在申请信用卡时，可以要求申请人提供驾驶证号，没有驾驶证的申请人自然使该字段为空。再如在做市场调查时常常会碰到有条件选择部分，仅当被调查者以特定方式回答前面问题时，条件部分才需要填写，但在存储时则可能会将所有的数据全部存储。

（3）有些信息是被遗漏的。造成遗漏的原因或是输入时忘记填写或对数据理解错误；或是数据采集设备、存储介质、传输媒体的故障；或是人为因素。

（4）有些信息（被认为）是不重要的。如一个属性的取值与给定语境是无关的，或训练数据库的设计者并不在乎某个属性的取值，如网络用户注册时许多信息是空缺的。

（5）要求统计的时间窗口并非对所有数据都适合。例如需要统计“客户在前 6 个月内的最大存款余额”，很明显，对于那些建立账户尚不满 6 个月的客户来说，统计出来的数值与想要得到的就可能存在差距。

（6）有些数据不符合格式要求。例如电话格式一般由三组数字组成，其中 3～4 个编号为区域代码，3～4 个编号为交换代码，4 个编号为节点。当某些记录不符合这些格式时，就可以认为是一种无效的记录，从而造成数据缺失。

在许多情况下，缺失值在数据源中用 NULL 表示。然而 NULL 有时是可接受的数值，在这种情况下，数值为空而不是缺失，它时常是下列两种意思之一：

- 没有充足的证据表明该字段对个体是否为真，例如没有订阅高尔夫球杂志意味着此人不打高尔夫，但不能证明。
- 重叠的数据中，对于此人没有与之匹配的记录。

区分这些情形是有用的。一种方法是分开记录不匹配的数据，创建两个不同的模型集；另一种方法是用另外的数值代替 NULL，指出匹配失败是在记录层次还是在字段层次。

常用的缺失数据处理方法有如下几种。

（1）忽略元组：当缺少类标号时，通常采用忽略元组的方法。但要注意除非元组中空缺值的属性较多，否则忽略元组不是有效的方法。

（2）忽略属性列：如果该属性的缺失值太多，如超过 80%，则在整个数据集中忽略该属性。

（3）人工填写空缺值：这个方法较为费时，尤其是数据集很大、缺少的值很多时，这个方法有可能行不通。

（4）数据填充：这类方法是用一定的值去填充缺失值，通常基于统计学原理，根据决策表中其余对象取值的分布情况来对一个空值进行填充。有三类不同的缺失值填充策略：

①用全局常量填充空缺值：用同一常数替换空缺的属性值。此方法虽然简单，但可能会对数据挖掘程序产生误导，根据填充的值可能得出有偏差甚至错误的结论，因此应谨慎使用。

②使用与给定记录属同一类的所有样本的均值或众数填充缺失值：假设某数据集的一条属于 a 类的记录在 A 属性上存在缺失值，那么可以用该属性上属于 a 类全部记录的平均值来代替该缺失值。如可用相同年级同学的平均年龄替换“年龄”属性中的空缺值。

③用可能值来代替缺失值：可以用回归分析、聚类、最近邻方法或决策树归纳确定最有可能的值填补空缺值。例如利用数据集中其他顾客的属性，可以构造一棵决策树来预测相同属性的缺失值；或是利用相互之间“接近”的对象具有相似的预测值预测其最近的邻居对象的缺失值，但要注意采用合适的距离定义，否则会产生较大的误差。

策略③是使用已有数据的大部分信息来预测缺失值，效果相对较好，但代价大；策略②实现起来简单、效率高，效果相对不错。但无论以哪种方式填充，都无法避免主观因素对原系统的影响，任何一个替换值会改变变量的分布，并且可能导致产生拙劣的模型。

（5）将模型拆分成几个部分。很多情况下，数据缺失是由于系统原因。较好的解决办法是将模型集拆分成几个部分，从一个数据集中消去缺失字段。虽然一个数据集中存在多个字段，但都不再有缺失值。例如考虑有 12 个月账单数据的客户标识特征，最好的办法是把模型集拆分成两部分，一部分模型集包含 12 个月保有期客户，另一个包含最近的客户。

19.2.2　消除噪声数据

噪声数据是指看起来正确但实际上不正确的属性值，噪声是测量变量的随机偏差，产生的原因有多种，可能是数据收集的设备故障，也可能是数据录入过程中人的疏忽或是数据传输过程中的错误等。

数据挖掘中使用的数据并不是为了数据挖掘而收集的。在最初收集数据时，数据的某些方面

可能并不重要，所以留下空白或没有被检查。这不会对收集数据的初衷造成影响，但当这些数据用于数据挖掘时，错误和省略的部分立刻变得相当重要。例如银行并不真正需要知道客户的年龄，所以它们的数据库中也许会存在许多缺失或不正确的年龄值，但在数据挖掘中得到的规则中，年龄有可能会是一种重要的特征。

噪声数据往往是离群值，所以很多情况下可以标识之。但有时噪声数据可能是隐含的模式。

一般可以采用以下方法消除之。

（1）分箱：分箱方法是通过考察“近邻”对象来平滑存储的数据值。平滑时可以按箱均值（箱中数据的均值代替每一个值）、箱中位数（用中位数替换箱中的每一个数据）、箱边界（最大或最小）值（将箱中的每一个值用与其接近的那个边值替换）。一般来说，宽度越大，平滑效果越好。箱可以是等宽的，每个箱的取值区间是一个常量。

（2）聚类：由于同一类具有高度的相似性，所以通过聚类可以发现离群点即噪声。删除离群点即可平滑数据。

（3）回归：可以用由数据拟合的函数来平滑数据。

（4）计算机与人工检查结合：可以通过计算机和人工检查结合的方法来识别孤立点。但要注意的是孤立点既可能是噪声，也可能包含有用的信息。计算机将差异程度大于阈值的模式记录到一个表中，通过审查表中的模式可以识别真正的噪声。

在噪声数据中，有两种极端的字段需要特殊处理：取值几乎相同和几乎都不同的字段。

只有一个取值或几乎只有一个取值的字段，包含的信息非常少，对于数据挖掘目的而言，应该忽略这些字段。但在忽略这些字段之前，应该了解为什么会出现如此倾斜的分布，它反映了对应于商业的何种事件。经验表明，如果列中 95%~99%的数值相同，在孤立情况下，如果不进行一些处理，该列可能毫无用处。

每一行或几乎每一行取不同值的分类属性字段（例如客户姓名、地址、电话号码、身份证号、学号和车牌号）虽然可唯一（或非常接近）识别每一行（每位客户），包含着丰富的信息（如学号包含了入学年份和专业信息），但它们不会在数据挖掘中被直接使用，这时需要借助领域知识从这些字段中提取重要特征作为衍生变量。

19.2.3 实现数据一致性

对于数据存在的不一致的数据，可以参照其他资料（如原始记录等）人为地加以更正，还可以使用用来纠正编码不一致问题的程序，也可以用知识工程工具来检测不符合条件约束的数据。

19.3 数据集成与转换

海量数据集往往涉及多个数据源，因此，在数据挖掘前需要合并这些数据源存储的数据。如果原始数据的形式不适合数据挖掘算法需要，就要对数据进行变换。

19.3.1 数据集成

数据集成是指将不同数据源的数据集中存放在一个统一的数据存储（如数据仓库）中。在集成的过程中由于语义上存在的差异会造成数据的不一致，并且存在冗余，需要进行消除。

数据不一致的原因是它们所指的是不同的对象。不同表中可能使用不同名称来指示同一属性；或者是不同的名称表示同一属性。数据不一致的另一种表现形式是数据值冲突，因为表示方法、比例或编码的不同，现实世界的同一实体在不同的数据库中的属性值可能不同。例如重量属性在一个系统中可能以公制单位存放，而在另一个系统中以英制单位存放；对于连锁旅馆，不同城市的房价不仅可能涉及不同货币，而且可能涉及不同的服务和税。

冗余是数据集成的另一个重要问题。如果一个属性能从另一个表“导出”，那这个属性就是冗余的。不一致的属性或伪命名也可以导致数据冗余。利用相关分析可以发现一些冗余的问题。

除了冗余外，在元组级还应当检测“重复”数据，即存在两个或多个相同的元组。

数据集成中涉及的重要问题是检测与处理冲突数据。解决冲突的简单办法是指定某一系统在冲突中占据主导地位。

19.3.2　数据转换

数据转换的目的是使数据和将来要建立的模型拟合得更好，形成适合挖掘的形式，它主要涉及以下的工作。

（1）平滑：可以采用分箱、聚类等方法去掉数据中的噪声。

（2）聚集：对数据进行汇总和聚集。

（3）数据概化：使用概念分层，用高层次概念替换低层次“原始”数据。

（4）规范化：数值规范化是将原来的度量转换为无量纲的值，是通过将属性数据按比例缩放，使之落入一个小的特定区间来规范属性。对于基于距离的方法，规范化可以帮助平衡具有较大初始值域的属性与具有较小初始值域人属性可比性。

规范化的方法有以下几种。

① 最小—最大规范化

最小—最大规范化对原始数据进行线性变换，保持原始数据值之间的线性关系，其计算公式为

$$x' = \frac{x - \min_A}{\max_A - \min_A}(new_\max_A - new_\min_A) + new_\min_A$$

式中：$\max_A$、$\min_A$、new_max_A、new_min_A 分别为原始及变换后属性值中的最大与最小值；x 为属性值。

最小—最大规范化保持原有数据之间的联系。如果今后的输入落在原始数据值域之外，该方法将面临“越界错误”。

② z-score 规范化（零均值规范化）

把属性 A 的值 x 基于 A 的均值和标准差规范化为 x'，由下式计算

$$x' = \frac{x - mean_A}{\sigma_A}$$

其中：mean_A 和 σ_A 分别为属性 A 的均值和标准差。在实际应用中，该方法常用于由于难以预知该属性的最大最小值，或者由于某些孤立点的存在的场合。

③ 小数定标规范化

通过移动属性 A 的小数点的位置进行规范化。小数点的移动位数依赖于 A 的最大绝对值。

将 A 的值 x 按下式公式规范化为 x'

$$x' = \frac{x}{10^j}$$

其中：j 是使 $\max(|x'|)<1$ 的最小整数。

经过规范化处理后，数据会发有很大的变化。因此要保留规范化参数，以便将来的数据可以用一致的方法规范化。

（5）属性构造（特征构造）：构造新属性并将其添加到属性集中有助于数据挖掘过程。例如要判断电信客户的消费倾向及忠诚度时，因为收集的原始特征集不可能直接包含这类，所以就需要进行构造。在人脸识别中，由于依照相片集合对人脸进行分类存在着许多困难，大量的分类算法都不合适。如果我们对相片数据进行处理，提供诸如某些类型的边和区域等与人脸高度相关的较高层次的特征，则更多的分类技术可以应用于该领域。

特征构造在不同的领域其应用方式不同。一旦数据挖掘用于一个相对较新的领域时，一个关键任务就是如何构造新的特征。特征的构造需要对领域知识和数据进行深入理解。

19.4 数据归约与压缩

19.4.1 数据归约

在数据挖掘的实践应用中，有许多情况需要对大数据进行科学的筛选和抽样，而不是对大数据本身直接应用数据挖掘算法。这主要是因为对大数据进行数据挖掘时间长、成本高，而且大数据的全集的数据质量并不能保证，混杂成分太多，主题特征不易清晰地辨别，直接影响数据挖掘的效果，所以需要对大数据集进行提炼，即数据归约。

在数据归约过程中，要注意归约后的数据要有代表性，要能代表数据总体的特征。为此，在归约过程要注意数据不能有偏、不能受到干扰以及产生偏移。

数据归约技术可以用来得到数据集的归约表示，虽然数据规模缩小了，但仍接近于原数据的完整性。这样，在归约后的数据集上进行挖掘效率更高，并能产生相同（或几乎相同）的分析结果。

常用的数据归约策略有以下几种。

1. 数据立方体

数据立方体可以存放多维聚集信息，图 19.1 即为某商店每类商品在各部门年销售多维数据，每个单元存入一个聚集值，对应于多维空间的一个数据点。最低层的数据立方体称为基本方体，最高层抽象的数据立方体称为顶点方体。用户感兴趣的是基本方体。

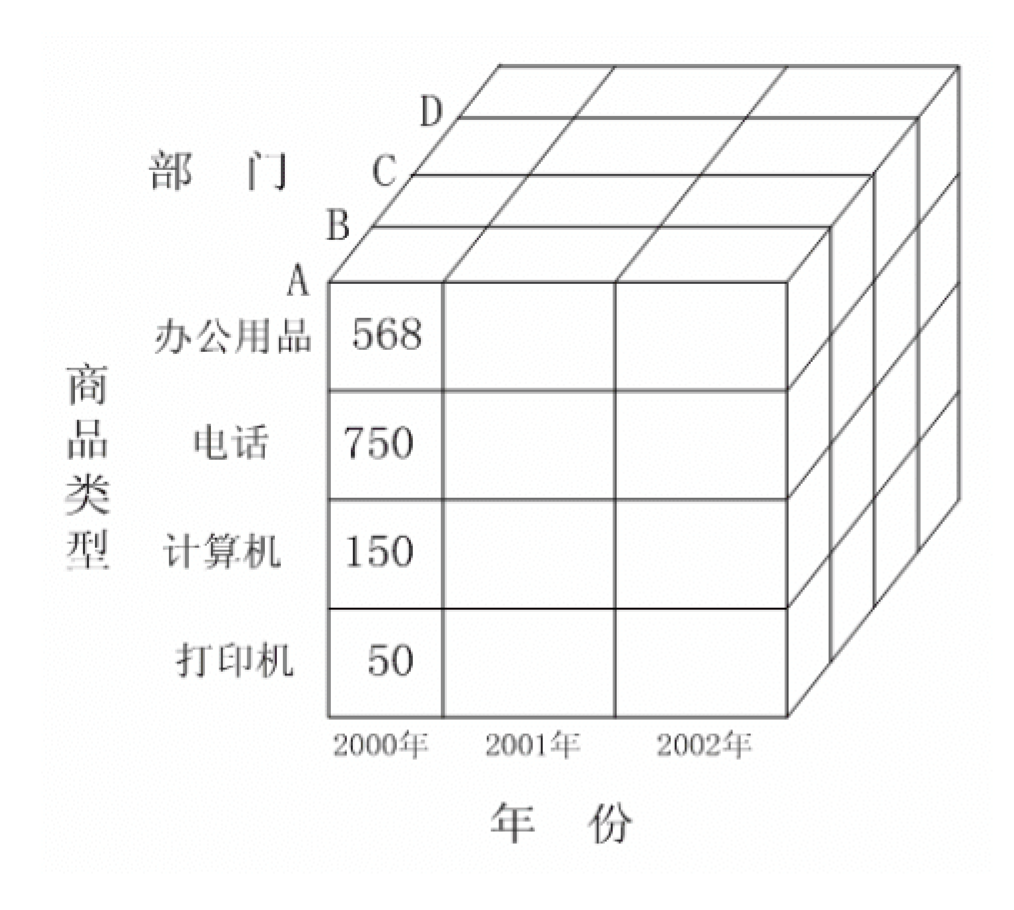


图 19.1 企业销售数据立方体

2. 维归约（特征选择）

用于数据分析的数据可能包含很多属性，其中一些属性与数据挖掘任务并不相关。不相关或冗余的属性增加了数据量，可能会减慢数据挖掘进程。

维归约通过删除与数据挖掘不相关的属性（维），达到减少数据量的目的。通常使用属性子集选择方法，找出最小属性集，使数据概率分布尽可能接近原始数据分布。通过维归约能减少模式上的属性数目，使模式更易于理解。

在实际应用中首先要尽量多列一些可能有影响的因素，然后通过数据处理，筛选出作用较大的特征，删除影响不大的特征，从而建立数学模型。特征筛选的第一步是分析每个特征，考察特征间的相关性，以及特征与目标相关性。各特征与目标值之间的相关系数为正时，其中 x_{ij} 和 y_i 分别表示第 i 个样品的第 j 特征值和目标值，$\overline{x}_j$ 和 $\overline{y}$ 分别表示第 j 个特征和所有样本目标值的均值，可以根据 $R(y,x_j)$ 绝对值的大小来判断各特征的重要性。

$$R(y,x_j)=\frac{\sum_{i=1}^{n}(y_i-\overline{y})(x_{ij}-\overline{x}_j)}{\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2\sum_{k=1}^{n}(x_{ij}-\overline{x}_j)^2}}$$

要注意的是对于相关系数小的特征，还需要用其他信息才能决定是否能删除。

尽管使用常识或领域知识可以消除一些不相关的或冗余的特征，但是选择最佳的特征子集通常需要系统的方法。特征选择的理想方法是将所有可能的特征子集作为感兴趣的数据算法的输入，然后选取产生最好结果的子集。这种方法的优点是反映了最终使用的数据挖掘算法的目的和偏爱。但由于子集数目太大（2^n 个），在大部分情况下这种方法行不通。

根据特征选择过程与后续数据挖掘算法的关联，特征选择方法可分为过滤、封装和嵌入。

- 过滤方法是指使用某种独立于数据挖掘任务的方法，在数据挖掘算法运行之前进行特征选择，即先过滤特征集和一个最有价值的特征子集。
- 封闭方法是将学习方法的结果作为特征子集评价准则的一部分，根据算法生成规则的分类精度选择特征子集。该类算法具有使得生成规则分类精度高的优点，但特征选择效率较低。
- 嵌入方法是将特征选择作为数据挖掘算法的一部分自然地出现。在数据挖掘算法运行期间，算法本身决定使用哪些属性和忽略哪些特征，如决策树 C4.5 分类算法。

根据是否用到类信息的指导，特征选择过程可分为监督式、无监督式和半监督式特征选择。

- 监督式特征选择使用类信息来进行指导，通过度量类信息与特征之间的相互关系来确定子集大小。
- 无监督式特征选择是在没有类信息的指导下，使用样本聚类或特征聚类对聚类过程中的特征贡献度进行评估，然后根据贡献度的大小进行特征选择。
- 半监督式特征选择是使用少量的有类信息的数据和无类信息的大量数据组合成数据集而进行特征选择。

特征选择过程一般由 4 部分组成：子集评估度量、控制新的特征子集产生的搜索策略、停止策略验证过程。

特征子集选择的策略主要包括以下技术。

- 逐步向前选择：以空属性集作为归约集开始，确定原属性集中最好的属性并将它添加到归约集中。在其后的每次迭代中，将剩下的原属性集中最好的属性添加到该集合中。
- 逐步向后删除：由整个属性集开始，在每一步删除尚在属性集中最差的属性。
- 向前选择和向后删除的结合：将逐步向前选择和向后删除方法结合在一起，每一步选择一个最好的属性，并在剩余属性中删除一个最差的属性。
- 决策树归纳：构造一个类似于流程图的结构，其中每个内部节点表示一个属性的测试，每个分支对应于测试的一个输出；每个外部节点表示一个类预测，在每个节点，算法选择“最好”的属性，将数据划分成类。如图 19.2 所示。

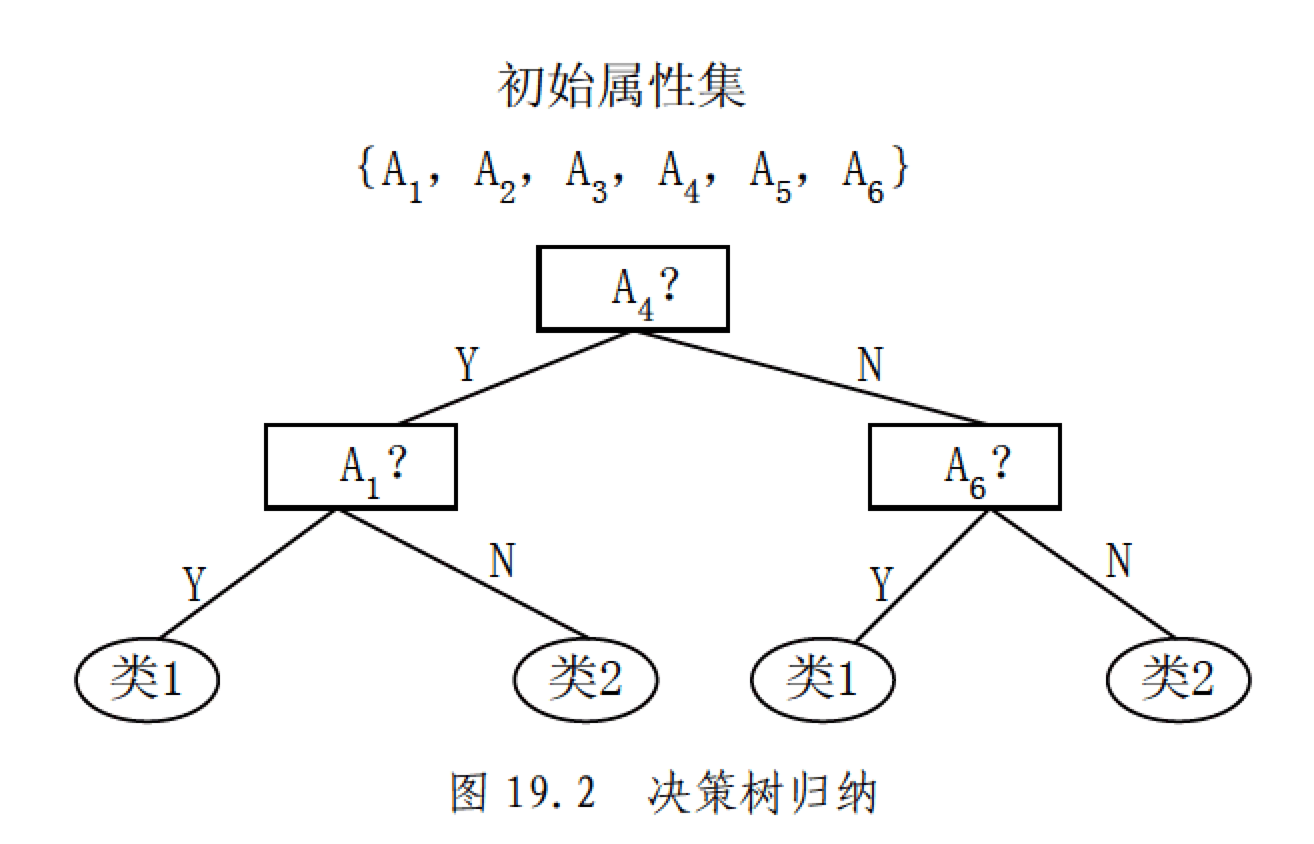


图 19.2 决策树归纳

在特征搜索过程中，一个不可或缺的环节是评估步骤，即与已经考虑的其他子集相比，评价当前的子集。评估策略需要一种评估度量以确定属性特征子集的质量。对于过滤方法，这种度量试图预测实际的数据挖掘算法在给定的属性集上执行的效果。常用的度量方法有相关度量、关联规则、粗糙集等。对于封装方法，评估包括实际运行目标数据应用，子集评估函数通常用于度量数据挖掘结果的标准。

评估的基础主要有三类：距离、概率密度函数和熵函数。

（1）基于分类误差的可分性判据。

一个理想的模式识别系统应能以最低的错误率分类未知模式。贝叶斯最小错误率决策的类概率误差计算公式由下式给出

$$e=\int[1-\max_{i} P(\omega_i \mid X)]P(X)\mathrm{d}X$$

其中：$P(\omega_i|X)$ 是第 i 类后验概率；$P(X)$ 是联合概率密度函数。但由于在一般情况下，误差不易计算，所以利用此方法提取特征难以实际进行。

（2）基于距离的可分性判据。

基于距离的可分性判据的出发点是各类样本间的距离越大，类内散度越小，则类别的可分性越好。令 $D(x_i,x_j)$ 为样本 i 与 j 之间的距离，则根据不同的定义，有欧氏距离、明考斯基（Minkoski）距离、马氏距离、切比雪夫距离等。

为了同时反映类内距离小和类间距离大的要求，可以构成准则函数

$$J_1 = \frac{J_b}{J_W} = \frac{tr(S_b)}{tr(S_W)}$$

式中：J_b、J_W分别为类间和类内总平均平方距离；S_b、S_W分别为类间和类内总散射矩阵；tr 为矩阵的迹。

由于 J_1 的值与坐标系统的选择有关，因此也可以采用以下的准则函数

$$J_2 = tr(S_W^{-1}S_b)$$
$$J_3 = \ln(S_W^{-1}S_b)$$
$$J_4 = tr(S_W^{-1}S_t)$$
$$J_3 = \ln(S_W^{-1}S_t)$$

式中：S_t 为所有样本之间的总平均平方距离。

（3）基于概率依赖度量的可分性判据。

模式向量 X 和类别 ω 的依赖性可以由条件概率密度函数 P（$\omega_i|X$）（i=1,2,$\cdots$,m）和联合概率密度 P（$\boldsymbol{X}$）之间的距离来度量。

Chernoff 距离：$Jc = -\ln\int P^s(X|\omega_1)P^{1-s}(X|\omega_2)\mathrm{d}X$

Bhattacharyya 距离：$J_B = -\ln\int P(X|\omega_1)P(X|\omega_2)^{1/2}\mathrm{d}X$

（4）基于熵度量的概率可分性判据。

熵的一般性定义为

$$J_E^\alpha = (2^{1-\alpha}-1)^{-1}\int[\sum_{i=1}^{M}P^\alpha(X|\omega_1)^{-1}]P(X)\mathrm{d}X$$

α 取不同值可以有不同的熵定义，如 $\alpha=1$ 称为 Shannon 熵，$\alpha=2$ 则得到平方熵。

与概率依赖度类似，熵度量也能估计模式向量 $\boldsymbol{X}$ 和类别 ω_i 之间的依赖性。

在大规模数据集中，由于特征数目很多，可能的子集数量也会很大，考察所有的子集可能不现实，因此需要某种停止搜索标准。其策略通常涉及一个或多个条件：迭代次数，子集评估的度量值是否最优或超过给定的阈值。一个特定大小的子集是否已经达到最优，其子集大小和评估标准是否同时达到最优，使用搜索策略的选择是否得到改进等，这些都是特征选择过程中需要考虑的问题。

特征的选择在数据挖掘中尽管研究很多，但尚无一通用的理论可以遵循。下列是几种常用的方法。

（1）偏差权重法。

对于分类而言，偏差大的变量比偏差小的变量更重要，特征的标准偏差为

$$S_k = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ik}-\bar{x}_k)^2}$$

式中：$\bar{x}_k$ 为 n 个样本的均值。

很明显同一类样本之间的方差即类内方差（$S_{j,I}$）较小，而类与类之间的方差即类间方差（$S_{j,o}$）较大，因此可定义权重因子

$$w_j = \frac{S_{j,O}}{S_{j,I}}$$

显然，w_j越大，特征j就越重要，应当优先选择。

（2）Fisher 比率法。

特征j的 Fisher 比率F_j为

$$F_j = \frac{\bar{x}_{j,1} - \bar{x}_{j,2}}{S_{j,1} + S_{j,2}}$$

式中：$\bar{x}_{j,1}$和$\bar{x}_{j,2}$分别为类 1 和类 2 中变量的均值；$S_{j,1}$和$S_{j,2}$分别是类 1 和类 2 中特征的标准偏差。F_j值越大，意味着此特征越重要，应优先选择。

（3）概率比率法。

概率比率R_j的定义为

$$R_j = \lg\frac{P_{j,1}}{P_{j,2}}$$

式中：$P_{j,1}$和$P_{j,2}$分别为第j个特征在类 1 和类 2 中出现的概率。根据此值的大小可判定：如果某特征在 2 类分类中均不出现或出现次数很少或出现概率相等，可以剔除。R_j绝对值越大，表明该特征量在同类中概率差最大，应优先选择。

（4）逐步判别法。

逐步判别分析为模式识别的一种方法，同时，该种方法亦用于变量的选择，特别是两变量共线，即相关系数较大时，用逐步判别的分析可以消去不合适的变量。

（5）学习机械法。

学习机械法也可以用于特征的选取。在特征选取时，首先将判别函数系数赋予任意初值，如均为“1”，然后，逐步校正，直到不能够进一步改善为止。再将值均赋予“-1”，重复上述过程，也直到不能够进一步改善为止。在两次结果中，剔除符号有改变的特征，重复上述全部过程，直到再无特征可剔除为止。

特征子集一旦选定，就需要根据数据挖掘任务进行目标验证，最直接的方法就是将特征全集的结果与该子集上得到的结果进行比较（一般从分类性能上进行比较）。如果理想的话，特征子集产生的结果将比使用特征全集产生的结果要好，或者几乎一样好。类似的验证方法还可以将不同特征选择算法得到的特征子集性能进行综合比较。

特征选择的算法有很多，下面即为一种非搜索型的特征选择方法（Fast Correlation_Based Filter，FCBF）。在这里利用互信息的方法来度量两个分类特征之间的相关性。用$P(x_i)$表示特征x取i个值x_i的概率，$P(x_i|y_j)$表示特征y取值为y_j时特征x取值为x_i的概率。x的信息熵$H(x)$及已知变量y后x的条件信息熵$H(x|y)$的计算方法如下

$$H(x) = -\sum_i P(x_i)\log_2 P(x_i)$$

$$H(x \mid y) = -\sum_j P(y_j)\sum_i P(x_i \mid y_j)\log_2 P(x_i \mid y_j)$$

变量x、y之间的互信息 MI(x,y)可按以下公式计算

$$\mathrm{MI}(x,y)=H(x)-H(x\mid y)=H(y)-H(y\mid x)=\sum_{x,y}P(xy)\log_2\frac{P(xy)}{P(x)P(y)}$$

用如下公式来度量特征 x 与特征 y 之间的相关性

$$\mathrm{Sim}(x,y)=\frac{2\mathrm{MI}(x,y)}{H(x)+H(y)}$$

采用FCBF 算法，求得每个特征 x_i 与目标特征 C（即类）的相关性 $\mathrm{Sim}(x_i,C)$，找出 $\mathrm{Sim}(x_i,C)\geqslant\delta$（阈值）的特征，然后再在这些特征中寻找那些较大相关性的特征（即支配性特征）直到删除所有的冗余特征。

19.4.2 数据压缩

通过对数据的压缩（也可以称作特征提取）可以把数据存储在很小的空间中。数据仓库尤其需要数据压缩，因为数据仓库中的数据很少更新。

特征提取作为一种特征空间维数压缩方法，其主要特点是在于通过变换的方法实现对原始特征的计算，使变换后的二次特征可以去掉一些分量（特征维数）。

对于 n 个原始特征构成的特征向量 $x=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$，特征提取就是对 x 作变换，产生 d 维向量 y= $(y_1,y_2,\cdots,y_d)^{\mathrm{T}}$，$d\leqslant n$，即

$$y=\boldsymbol{W}^{\mathrm{T}}x$$

其中：$\boldsymbol{W}=\boldsymbol{W}_{n\times d}$ 称为特征提取矩阵或简称变换矩阵。基于可分性判据的特征提取就是在一定的可分性判据下，如何求最优的变换矩阵 $\boldsymbol{W}$。

如果可以不丢失任何信息地还原压缩数据，那么使用的数据压缩技术就是无损的，相反就是有损的。主成分分析及小波变换等方法是常用的无损的数据压缩技术。

19.4.3 数值归约

数值归约技术利用替代数据以“较小的”数据表示形式来达到减少数据量的目的，其常用的方法如下。

1. 回归和对数线性模型

回归是研究自变量与因变量之间关系的分析，目的在于根据已知自变量来估计和预测因变量的总平均值。回归和对数线性模型可以近似拟合给定的数据。线性回归是最简单的回归形式。对数线性模型可用于估计具有离散属性值的基本方体中每个格的概率分布。该模型允许由较低阶的数据立方体构造较高阶的数据立方体。

2. 直方图

数据总结的最好方法是提供数据的直方图，可以从中获得对数据的更高层次的理解。无论对于近似稀疏、稠密数据、高倾斜数据或一致的数据，直方图都是一种有效的方法。

3. 聚类

在数据归约时，用数据的聚类表示替换实际数据。如果数据能够组织成不同的聚类，聚类技术是一种很有效的方法。在保证样本代表整个数据集的前提下，在样本数据上应用数据挖掘算法，显然比直接在整个数据集上进行数据挖掘效率更高。

聚类的“质量”可以用“直径”或“质心”表示，直径表示一个聚类中任意两个对象间的最大距离；质心表示由聚类中心到每个聚类对象的平均距离。

4. 抽样

抽样也可以作为一种数据归约技术，它用较小的随机样本（子集）表示大的数据集。假设海量数据 D 包含 N 个对象。可以用如下方法对 D 抽样。

（1）不放回简单随机抽样：从数据集 D 中的 N 个对象中逐个不放回地抽取 n 个（$n<N$），抽取 D 中任何对象的概率均为 $1/N$。一个对象一旦被抽取，就不可能再被抽到。

（2）放回简单随机抽样：从 N 个对象中抽取一个对象，每次抽取时各对象被抽取的概率为 $1/N$，将抽到的对象记录后再放回总体，重复上述过程 n 次。很明显，一个对象有可能被多次抽到。

由于数据挖掘时所面临的是海量数据，因此上述两个抽样方法的差异可以忽略。

（3）整群抽样：抽样的单位不是单个的个体，而是成群的个体。它是从总体中随机抽取一些小的群体，然后由这些小群体内的所有元素构成调查样本。对小群体的抽取可以采用简单随机抽样、系统抽样和分层抽样等方法。

整群抽样方法简便易行，节省费用，非常适合在难以确定总体抽样的情况。但如果样本分布比较集中，此方法的代表性较差。

（4）分层抽样：把总体分成不重叠的层，从每一层分别抽取样本，然后由各层子样本组成总体的样本。

分层抽样是一种常用的抽样技术，它不仅可以对总体目标量进行估计，也可以对各层的目标量进行估计。例如可以对一个顾客数据集按照年龄进行分层，再在每个年龄组中进行随机选择，从而确保了最终获得分成抽样数据子集中的年龄分布具有代表性。

（5）多阶抽样：按照元素的隶属关系和层次关系，把抽样过程分为几个阶段进行。适用于总体规模特别大，或者总体分布范围特别广的情况。但此方法会产生误差。可以通过增加初始阶段的样本数，适当地减少末尾阶段的样本数来减少误差。

（6）系统抽样：将总体中的对象按某种顺序排列，在规定的范围内随机抽取一个或一组对象，然后按一定规则确定其他样本对象。

19.5 数值数据的概念分层与离散化

19.5.1 概念分层

概念分层（简称概化）定义了一组由低层概念集到高层概念集的映射。它允许在各种抽象级别上处理数据，从而在多个抽象层上发现知识。

概念分层结构可以用树来表示，树的每个节点代表一个概念。树根节点表示给定维的最一般的值。通常，概念分层结构中的层自顶向下编号，树根节点为 0 层，其余类推。如图 19.3 所示。

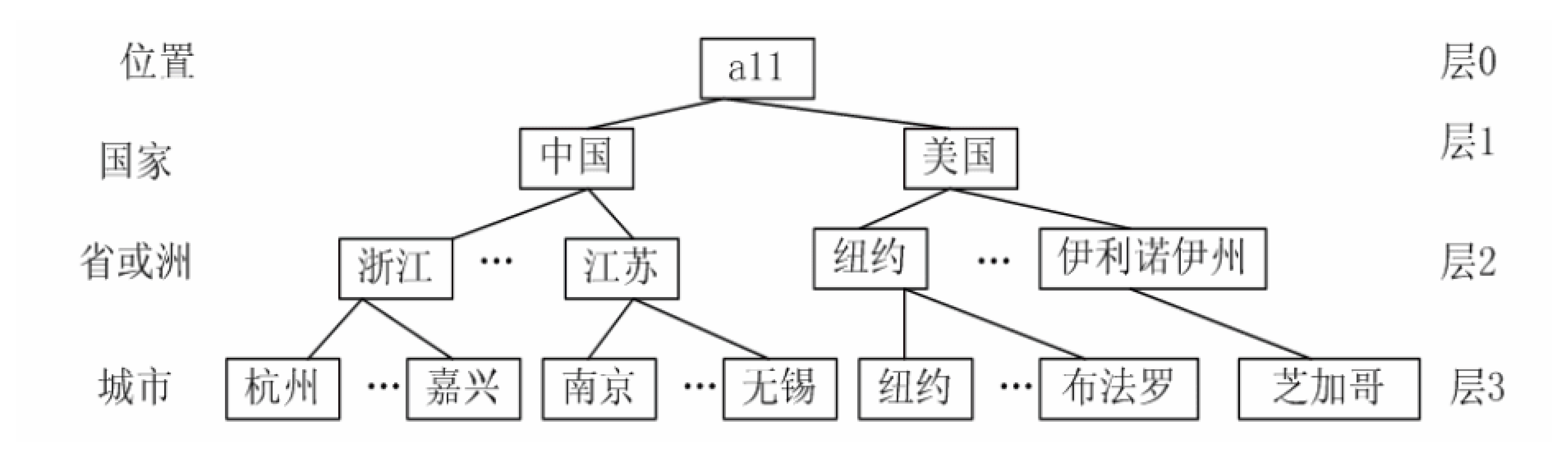


图 19.3　“位置”维的概念分层

通过数据概化可以让用户在更有意义、更清晰的抽象层观察数据，从中发现更易于理解的模式；也可以压缩数据，在压缩的数据集上进行数据挖掘更为有效。

19.5.2　概念分层的类型

概念分层包括模式分层、集合分组分层、由操作导出的分层和基于规则的分层等类型。

1. 模式分层

模式分层是数据库模式属性间的全序或偏序。通常情况下，一个模式分层指定数据仓库的一个维，维的属性也可以组织成偏序，形成一个格。如图 19.4 所示就是关于时间的分层。

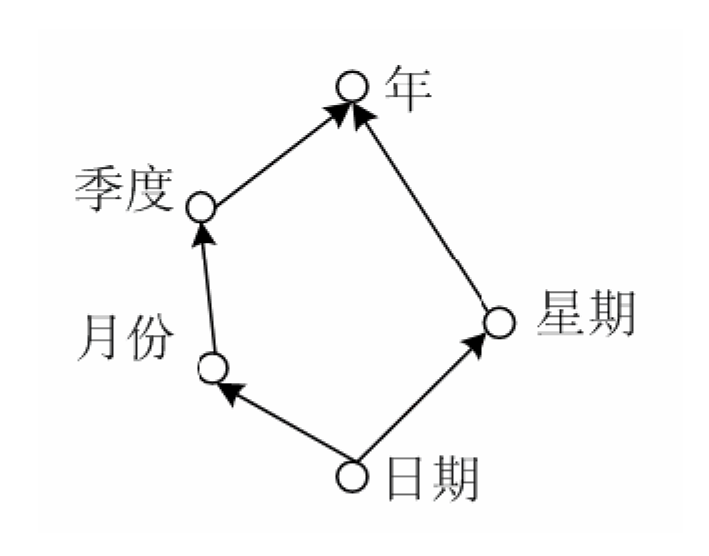


图 19.4　概念分层的格结构

2. 集合分组分层

集合分组分层将给定属性或维的值组织成常量组或区间，也就是通过维或属性值的离散化或分组来定义分层。组之间可以定义或偏序。当两种类型的分层结构结合时，集合分组分层可以用于精练或丰富模式定义的分层。

3. 由操作导出的分层

由操作导出的分层是根据用户、专家或数据挖掘系统指明的操作分层。操作可能包括对信息编码串的解码，从复杂数据对象提取信息和数据聚类等。

例如，从一个 E-mail 地址 aa@cs.jlu.cn 中发现偏序“用户名＜院系＜学校＜国家”，形成了 E-mail 地址的一个概念分层。

4. 基于规则的分层

基于规则的分层是指用一组规则定义整个概念分层或概念分层的某一部分，可以根据当前数据库和规则定义动态地计算分层。

19.5.3 数值数据离散化

数据离散化即将连续性数据（数值数据）转换成离散性数据。由于数据挖掘算法只能应用于离散型数据，使得离散化处理成为必要，而且离散化的结果将会减少给定连续变量值的个数，减少和简化原来的数据。

离散化处理一般经过以下几个步骤。

（1）对变量进行排序。

（2）选择某个点作为候选断点，根据给定的条件，判断此断点是否满足要求。

（3）若候选断点满足离散化的要求，则对数据集进行分裂或合并，再选择下一个候选点。

（4）重复以上步骤，直至满足停止准则，从而得到最终的离散结果。

可以根据对数据分布的统计分析构造数据属性的概念分层，在此基础上对数据离散化。常用的方法有分箱、直方图分析、聚类分析、基于熵的离散化和通过“自然划分”的数据分段等。

数据平滑的分箱方法也是一种数值离散化的方法。通过将数据分布到箱中，并用箱中数据的均值或中位数替换箱中的每个值，可以将属性值离散化。不断用这个方法划分结果，就能产生概念分层。

直方图也可以用于数据离散化。在等宽直方图中，将值划分成相等的部分或区间。在等深直方图中，对值进行划分使每一部分包括相同数目的样本。

聚类分析算法将数据划分成若干个簇。每个簇形成概念层的一个节点，所有的节点在同一概念层。将每个簇进一步分成若干子簇，形成较低的概念层。子簇聚集在一起，就形成较高的概念层。

熵是信息学中的一种度量，用来递归地划分数据属性，使之分层离散化。给定一个数据元组的集合 S，基于熵的概念对属性 A 离散化的方法如下。

（1）A 的每个值是一个潜在的区间边界或阈值 T。例如 A 的值 x 可以将样本 S 划分成分别满足条件 $A<x$ 和 $A\geqslant x$ 的两个子集，从而实现一个二元离散化。

（2）给定 S，选择在划分后信息增益最大的值作为阈值。

设 S_1 和 S_2 分别对应于 S 中满足条件 $A<T$ 和 $A\geqslant T$ 的样本，给定 m 个类，p_i 是类 i 在 S_1 中的概率，其值为 S_1 中含类 i 的样本数除以 S_1 中的样本总数。S_1 的熵函数 Ent 的定义如下

$$\mathrm{Ent}(S_1) = -\sum_{i=1}^{m} p_i \log(p_i)$$

$\mathrm{Ent}(S_2)$的值的计算与上类似。

划分的信息增益定义为

$$I(S,T) = \frac{|S_1|}{|S|}\mathrm{Ent}(S_1) + \frac{|S_1|}{|S|}\mathrm{Ent}(S_1)$$

选择信息增益较大的 T 作为阈值。

（3）把确定阈值的过程递归地用于每个划分，直至满足某个终止条件为止，例如

$$\mathrm{Ent}(S)-I(S,T)>\delta$$

基于熵的离散化可以压缩数据量。由于使用类信息，就更有可能将区间边界定义在准确位置，有助于提高分类的准确性。

通过自然划分分段也可以使概念分层。

19.5.4　分类数据的概念分层

分类数据是离散数据。一个分类属性具有有限个取值，值之间是无序的。针对分类数据的概念分层方法有如下几种。

（1）由用户或领域专家在模式级给出属性的部分序：分类属性或维的概念分层涉及一组属性。由用户或专家在模式级给出属性的部分序或全序，可以很容易地定义概念分层。

（2）通过显层数据分组给出分层结构：这是人工定义概念分层结构。

（3）只说明属性集，不说明它们的偏序：用户可以说明一个属性集，形成概念分层，但并不显示说明它们的偏序。系统自动地产生属性的序，构造有意义的概念分层。

（4）只说明部分属性值：在定义分层时，用户可能只说明了相关属性的一小部分。为了处理这种部分说明的分层结构，有必要在数据库模式中嵌入数据语义，把语义密切相关的属性捆绑在一起。这样一个属性的说明可以触发整个语义相关的属性组，从而形成一个完整的分层结构。

19.6　例题

例 4.1　某公司对应聘人员进行能力测试，测试成绩如表 19.1 所示。计算数据集的相应统计量。

表 19.1　应聘人员测试成绩

64	67	70	72	74	76	76	79	80	81
82	82	83	85	86	88	91	91	92	93
93	93	95	95	95	97	97	99	100	100
102	104	106	106	107	108	108	112	112	114
116	118	119	119	122	123	125	126	128	133

解：

各统计量计算如下：

```
>> x =[64 67 70 72 74 76 76 79 80 81;82 82 83 85 86 88 91 91 92 93;
      93 93 95 95 95 97 97 99 100 100;102 104 106 106 107 108 108
      112 112 114;116 118 119 119 122 123 125 126 128 133];
>> dts(x)
均值：97.68  方差：306.9567  标准差：17.5202  极差：69  变异系数：17.9363
偏度：0.087711  峰度：2.1895
>> fws(x)
```

```
中位数：96  下四分位数：83  上四分位数：112  四分位极差：29
三均值：96.75  下截断点：39.5  上截断点：155.5
>>qqs(x)            %QQ 图（图 19.5）
>> sfpin(x)         %频率直方图（图 19.6）
>> xxt(x)           %盒形图（图 19.7）
>> scdfplot(x)      %经验分布函数（图 19.8）
```

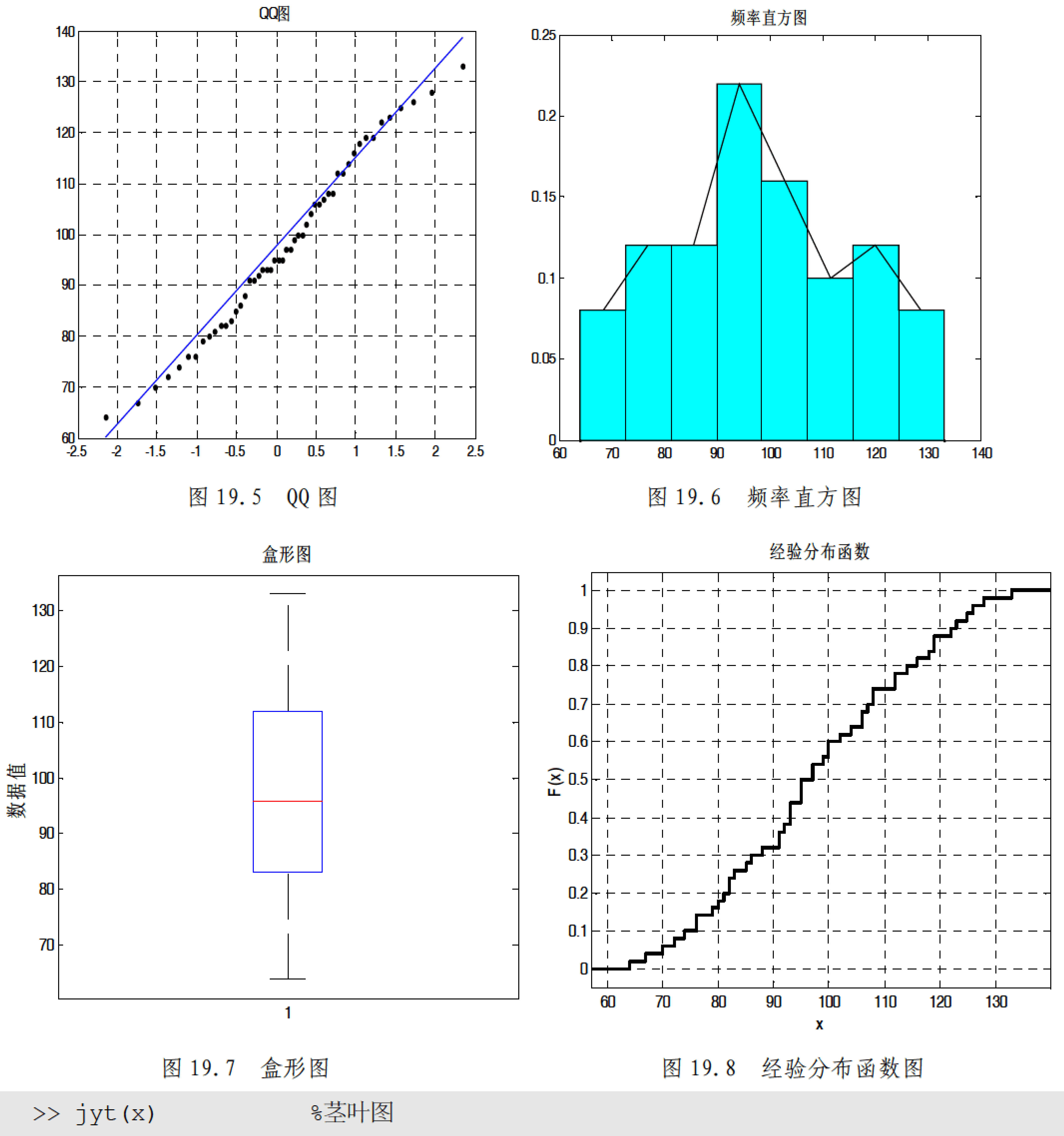


图 19.5 QQ 图

图 19.6 频率直方图

图 19.7 盒形图

图 19.8 经验分布函数图

```
>> jyt(x)           %茎叶图
60 : 4 7
70 : 0 2 4 6 6 9
80 : 0 1 2 2 3 5 6 8
90 : 1 1 2 3 3 3 5 5 5 7 7 9
```

```
100 : 0 0 2 4 6 6 7 8 8
110 : 2 2 4 6 8 9 9
120 : 2 3 5 6 8
130 : 3
```

例 4.2　对下列数据进行规范化处理。

```
data=[0.0390  0.9800  46.2000  6.3200; 0.0510  0.5800  32.9000  4.8500;
      0.0090  0.8000  50.9000  6.4800; 0.0420  0.9200  55.5000  6.2700;
      0.0260  1.5600  43.2000  5.4500; 0.0340  0.7400  59.2000  7.1300;
      0.0160  0.7500  41.6000  4.5600; 0.0190  0.8200  33.2000  7.0600;
      0.0370  0.9400  36.8000  6.2100; 0.0510  0.8700  33.7000  6.1700;
      0.0710  1.1300  31.4000  7.1900; 0.0550  0.8700  35.9000  5.5300];
```

解：

```
>> data1=guiyi(data,1);                 %归一化函数，求z-score规范化
>> data2=guiyi_range(data,[0 1])        %最小－最大规范化
```

例 4.3　图 19.9 为一模拟信号图，试用小波分析对其进行解析。

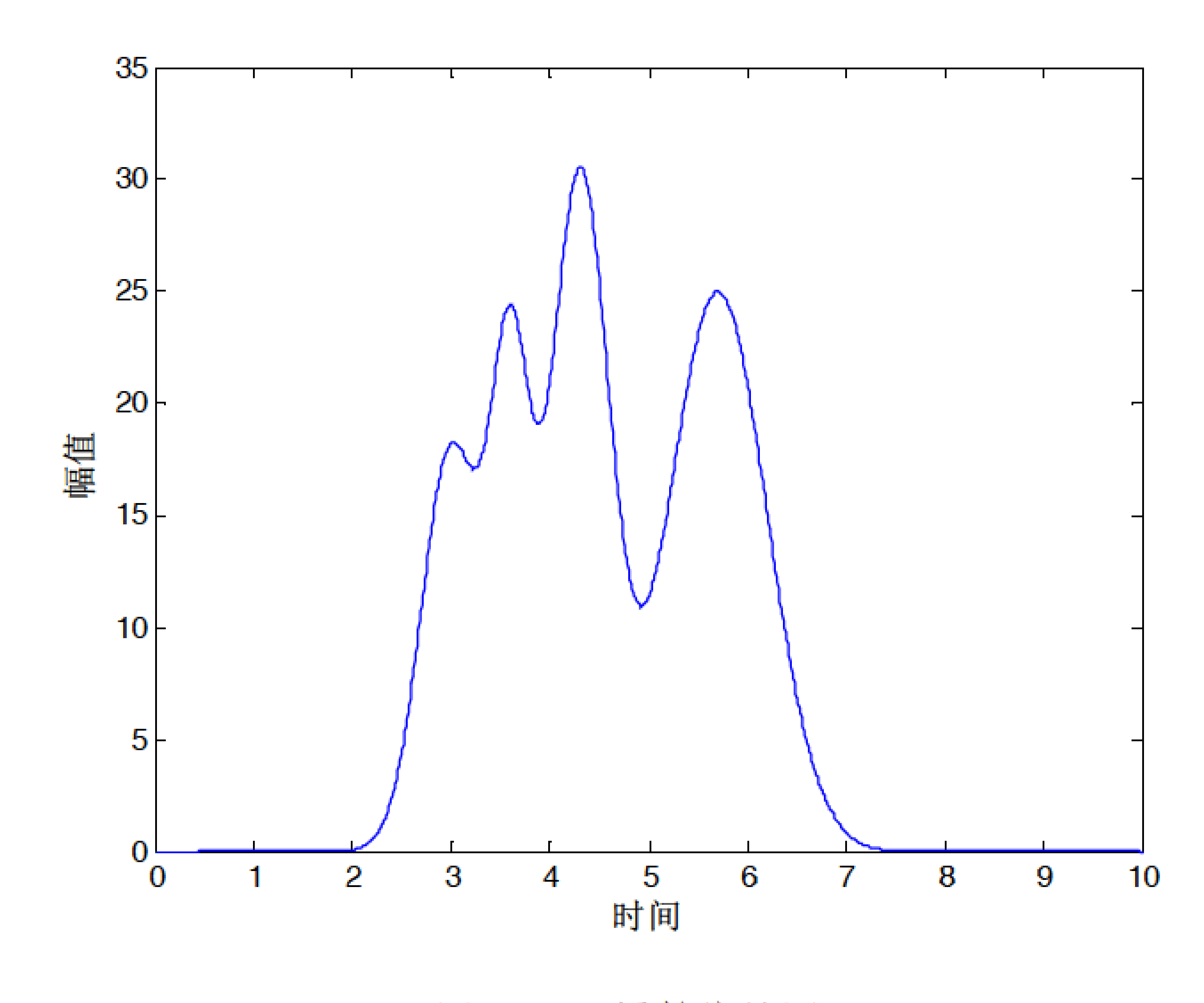


图 19.9　原始信号图

解：

```
>> x=linspace(0,10,1000);
>>y=25*exp(-(x-5.7).^2/(2*0.5^2))+30*exp(-(x-4.3).^2/(2*0.3^2))+20*exp(-
   (x-3.6).^2/(2*0.2^2))+18*exp(-(x-3.0).^2/(2*0.3^2));        %原始信号模拟
>> [c,l]=wavedec(y,7,'sym6');                                   %小波分解
>> d7=wrcoef('d',c,l,'sym6',7);
```

```
>> d6=wrcoef('d',c,l,'sym6',6);
>> d5=wrcoef('d',c,l,'sym6',5);
```

根据小波分析，不同尺度下的近似系数和细节系数代表着原始信号中的不同频率成分。最高频率的成分往往是噪声信号，最低频率的成分往往是基线或背景信号，而频率介于噪声和基线的成分则代表了信号的有用信息。

从计算结果可知，小波分解得的 d6 细节系数基本上能代表原图中的有用信息（如图 19.10 所示），可以用它作进一步的定量或定性的。

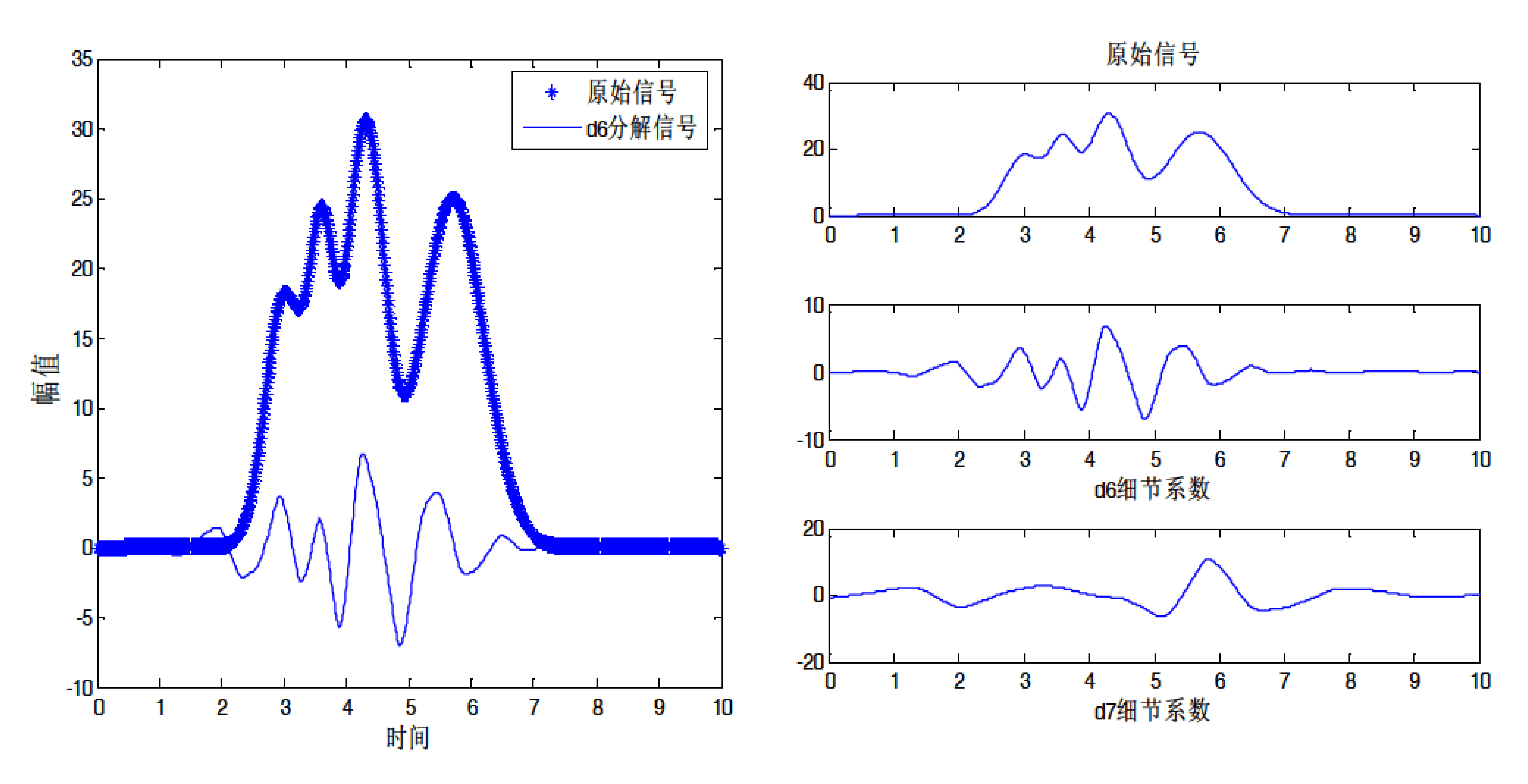


图 19.10 计算结果图

例 4.4 对例 4.2 中的数据利用主成分分析进行降维处理。

解：

```
>>data=[0.0390  0.9800  46.2000  6.3200; 0.0510  0.5800  32.9000  4.8500;
        0.0090  0.8000  50.9000  6.4800; 0.0420  0.9200  55.5000  6.2700;
        0.0260  1.5600  43.2000  5.4500; 0.0340  0.7400  59.2000  7.1300;
        0.0160  0.7500  41.6000  4.5600; 0.0190  0.8200  33.2000  7.0600;
        0.0370  0.9400  36.8000  6.2100; 0.0510  0.8700  33.7000  6.1700;
        0.0710  1.1300  31.4000  7.1900; 0.0550  0.8700  35.9000  5.5300];
>> y=myprincomp1(sample);     %可得到三个主成分，也即降了一维
```

例 4.5 特征变量的选择可以用多种方法，其中 ReliefF 算法是较为常用的方法。它是根据各个特征和类别的相关性赋予特征不同的权重，权重小于某个阈值的特征将被移除。算法从训练集 D 中随机选择一个样本 R，然后从和 R 同类的样本中寻找最近邻样本 H，称为 Near Hit，从和 R 不同类的样本中寻找最近邻样本 M，称为 NearMiss，然后根据以下规则更新每个特征的权重

$$W(A)=W(A)-\frac{\sum_{j=1}^{k}\mathrm{diff}(A,R,H_j)}{\mathrm{mk}}+\sum_{C\notin\mathrm{class}(R)}[\frac{p(C)}{1-p(\mathrm{class}(R))}\sum_{j=1}^{k}\mathrm{diff}(A,R,M_j(C))]/\mathrm{mk}$$

式中：diff(A,R_1,R_2)表示样本 R_1 和样本 R_2 在特征 A 上的差，其计算如下：

$$\mathrm{diff}(A,R_1,R_2)=\begin{cases}\dfrac{|R_1[A]-R_2[A]|}{\max(A)-\min(A)} & \text{if A is continuous}\\ 0 & \text{if A is discrete and } R_1[A]=R_2[A]\\ 1 & \text{if A is discrete and } R_1[A]\neq R_2[A]\end{cases}$$

特征的权重越大，表示该特征的分类能力越强；反之，表示该特征分类能力越弱。ReliefF 算法的运行时间随着样本的抽样次数 m 和原始特征个数 N 的增加线性增加，因而运行效率非常高。

利用此算法对来自 UCI 机器学习数据库（http://archive.ics.uci.edu/ml/）中的 Breast Cancer Wisconsin (Original) Data Set（威斯康星州乳腺癌数据集）进行变量选择。

解：

```
>> clear
>> a=dlmread('D:\数据 1.txt');D=a(:,2:end-1);class_L=a(:,end); k=8;m=80;
>> typeD=[0  0  0  0  0  0  0  0  0];
>> W=reliefF(D,class_L,m,k,typeD)    %可以多次求解,然后求平均值
W=0.2323  0.2153  0.1872  0.1396  0.1236  0.1163  0.1075  0.0536  0.0497  %权重
  6.0000  8.0000  1.0000  3.0000  7.0000  2.0000  4.0000  5.0000  9.0000 %特征序列
```

按照从小到大的顺序排列，可知，各个属性的权重关系如下：

```
属性 9<属性 5<属性 7<属性 4<属性 2<属性 3<属性 8<属性 1<属性 6
```

从上面的特征权重可以看出，属性 6 裸核大小是最主要的影响因素，说明乳腺癌患者的症状最先表现了裸核大小上，将直接导致裸核大小的变化，其次是属性 1 和属性 8 等，后几个属性权重大小接近，但是从多次计算规律来看，还是能够说明其中不同的重要程度。

例 4.6　非负矩阵因子分解算法（Non-negative factorization，NMF）是一种特征抽取算法。它常常用于属性很多、且属性模糊或者有很弱的可预测性的数据集。其基本思想是将数据矩阵 $\boldsymbol{V}$ 压缩为两个低级矩阵 $\boldsymbol{W}$ 和 $\boldsymbol{H}$ 乘积，使其近似等于 $\boldsymbol{V}$，即 $\boldsymbol{V}\approx\boldsymbol{WH}$。算法中使用交互式过程来修改矩阵的初始值（它可以是任意的），以便这个乘积接近矩阵 $\boldsymbol{V}$。迭代过程在达到一定的次数或者目标函数达到一定的误差内结束。

NMF 算法收敛速度快，占用存储空间少、分解结果可解释，能将高维的数据矩阵降维，适合处理大规模数据，已在图像处理、文本挖掘等领域得到广泛应用。

试利用 NMF 算法对 UCI 机器学习数据库中的 wine 数据集进行分解处理。

解：

NMF 算法的迭代公式如下：

$$W\leftarrow W\frac{(XH^T)}{(WHH^T)}$$

$$H\leftarrow H\frac{(W^TH)}{(W^TWH)}$$

目标函数为

$$\|V-WH\|^2=\sum_{ij}(V_{ij}-(WH)_{ij})^2$$

当且仅当 $\boldsymbol{V}=\boldsymbol{WH}$ 时目标函数为 0，得到近似分解的最优解。

据此可编程计算如下，算法中利用主成分分析确定 $\boldsymbol{W}$ 和 $\boldsymbol{H}$ 的初始矩阵，以及 r 数。

```
>> a=dlmread('D:\wine.txt');
>> x=a(:,2:end);
[w,h]=nmf(x,8000);
```

迭代 5000 次左右，便可以得到一个收敛的结果，其中 r 的数值为 6，即 w 为 13×6 的矩阵。

事实上，MATLAB 2008a 的版本中有此函数。

例 4.7　在属性选择中，根据属性的重要性来进行选择也是一种常用的方法。属性重要根据输入输出关联法进行计算，计算方法如下：

$$C(k)=\sum_{i\neq j}|x(i,k)-x(j,k)|\times sign(y(i)-y(j))$$

式中：$x(i,k)$、$x(j,k)$为 i、j 个样本的归一化后的第 k 个属性值；$y(i)$、$y(j)$为对应的目标值。

试利用此法对 Iris 数据进行分析。

解：

根据此法的原理，编程计算如下。

```
>> a=dlmread('D:\数据.txt'); x=a(:,1:4);
>> class=[ones(50,1);2*ones(50,1);3*ones(50,1)];
>> y=important(x,class);
y =1.0e+003 *
    2.3555    1.6891    3.4683    3.7119
```

从计算结果可看出，属性重要性的排序顺序为：4　3　1　2，与其他方法确定的顺序完全一致。

属性按重要性排序后，便可以进行属性的选择：

（1）取最重要的前 $n/2$ 或（$n+1$）/2 个属性进行模型的训练和验证，如果效果很好，依次去掉一个所选属性中最不重要的一个属性重新计算，如此重复计算，直至效果不好为止，如果第 k 次检测效果开始变得不好，则最终可以取 $k+1$ 个属性作为较优属性参与模型训练及验证。

（2）相反，如果一开始效果就不好，则依次加入未选属性中最重要的一个，直至效果变好为止。

例 4.8　相关分析在属性选择中得到了广泛应用，它主要是通过计算各类相关系数、χ^2 检验等过程来推断变量间是否存在相关关系。

一种原料来自三个不同的地区，原料质量被分成三个不同等级，从这批原料中随机抽取 500 件进行检验，得到如表 19.2 所示的结果。试在显著性水平 $\alpha=0.05$ 下，说明地区与原料间是否存在相关关系，如有，则关系的强弱如何？

表 19.2　不同地区的原料质量检测结果

	甲 地 区	乙 地 区	丙 地 区
一级	52	60	50
二级	64	59	65
三级	24	52	74

解：

此类问题即为独立性检验，是 χ^2 检验的另一种检验方式，它用来检验两个变量间是否存在联系的问题。

此题检验问题的原假设为

$$H_0: p_{ij} = p_{i\cdot}p_{\cdot j}, \quad \forall i,j, 1 \leqslant i \leqslant r, 1 \leqslant i \leqslant s,$$

式中：r 和 s 分别为变量 X 与 Y 的等级。即如果变量 X 与 Y 是独立的，或者说是没有关系，则 X 和 Y 的联合概率应该等于 X 和 Y 的边缘概率之积。

则根据相应统计的原理，可编程计算如下。

```
>> x=[52 60 50;64 59 65;24 52 74];
>> [p,h,para]=mychi2(x,'s');
>> p=5.4135e-004           %概率值
>> b=1                     %拒绝原假设，即认为地区和原料等级间存在依赖关系
>>para=                    %从各相关系数可看出，产地与原料等级间的相关程度不同
    ph: 0.1991             %ψ 相关系数
     C: 0.1953             %列联相关系数
     V: 0.1408             %V 相关系数
```

例 4.9　某公司在一项议案的调查中，得到如表 19.3 所示的列联表。问哪些因素与态度有关？哪些因素与态度无关？

表 19.3　调查数据

态度（Z）	支 持		反 对		弃 权	
工种（Y）	蓝领	白领	蓝领	白领	蓝领	白领
性别（X）						
男	60	50	95	40	34	44
女	80	45	105	41	44	53

解：

此例题涉及对数线性模型。根据对数线性模型编程计算如下。

```
>> x1={[60 50;80 45]};x2={[95 40;105 41]};x3={[34 45;44 53]};
>> x=[x1 x2 x3];
>> [a,b,c,d]=mychi2(x,'m');
```

表 19.2　不同地区的原料质量检测结果

	甲 地 区	乙 地 区	丙 地 区
一级	52	60	50
二级	64	59	65
三级	24	52	74

解：

此类问题即为独立性检验，是 χ^2 检验的另一种检验方式，它用来检验两个变量间是否存在联系的问题。

此题检验问题的原假设为

$$H_0: p_{ij} = p_{i\cdot} p_{\cdot j}, \quad \forall i, j, 1 \leqslant i \leqslant r, 1 \leqslant i \leqslant s,$$

式中：r 和 s 分别为变量 X 与 Y 的等级。即如果变量 X 与 Y 是独立的，或者说是没有关系，则 X 和 Y 的联合概率应该等于 X 和 Y 的边缘概率之积。

则根据相应统计的原理，可编程计算如下。

```
>> x=[52 60 50;64 59 65;24 52 74];
>> [p,h,para]=mychi2(x,'s');
>> p=5.4135e-004          %概率值
>> b=1                    %拒绝原假设，即认为地区和原料等级间存在依赖关系
>>para=                   %从各相关系数可看出，产地与原料等级间的相关程度不同
    ph: 0.1991            %ψ 相关系数
     C: 0.1953            %列联相关系数
     V: 0.1408            %V 相关系数
```

例 4.9　某公司在一项议案的调查中，得到如表 19.3 所示的列联表。问哪些因素与态度有关？哪些因素与态度无关？

表 19.3　调查数据

态度（Z）	支　持		反　对		弃　权	
工种（Y）	蓝领	白领	蓝领	白领	蓝领	白领
性别（X）						
男	60	50	95	40	34	44
女	80	45	105	41	44	53

解：

此例题涉及对数线性模型。根据对数线性模型编程计算如下。

```
>> x1={[60 50;80 45]};x2={[95 40;105 41]};x3={[34 45;44 53]};
>> x=[x1 x2 x3];
>> [a,b,c,d]=mychi2(x,'m');
```

表 19.2　不同地区的原料质量检测结果

	甲 地 区	乙 地 区	丙 地 区
一级	52	60	50
二级	64	59	65
三级	24	52	74

解：

此类问题即为独立性检验，是 χ^2 检验的另一种检验方式，它用来检验两个变量间是否存在联系的问题。

此题检验问题的原假设为

$$H_0: p_{ij} = p_{i\cdot}p_{\cdot j}, \quad \forall i, j, 1 \leqslant i \leqslant r, 1 \leqslant i \leqslant s,$$

式中：r 和 s 分别为变量 X 与 Y 的等级。即如果变量 X 与 Y 是独立的，或者说是没有关系，则 X 和 Y 的联合概率应该等于 X 和 Y 的边缘概率之积。

则根据相应统计的原理，可编程计算如下。

```
>> x=[52 60 50;64 59 65;24 52 74];
>> [p,h,para]=mychi2(x,'s');
>> p=5.4135e-004          %概率值
>> b=1                    %拒绝原假设，即认为地区和原料等级间存在依赖关系
>>para=                   %从各相关系数可看出，产地与原料等级间的相关程度不同
    ph: 0.1991            %ψ 相关系数
     C: 0.1953            %列联相关系数
     V: 0.1408            %V 相关系数
```

例 4.9　某公司在一项议案的调查中，得到如表 19.3 所示的列联表。问哪些因素与态度有关？哪些因素与态度无关？

表 19.3　调查数据

态度（Z）	支　持		反　对		弃　权	
工种（Y）	蓝领	白领	蓝领	白领	蓝领	白领
性别（X）						
男	60	50	95	40	34	44
女	80	45	105	41	44	53

解：

此例题涉及对数线性模型。根据对数线性模型编程计算如下。

```
>> x1={[60 50;80 45]};x2={[95 40;105 41]};x3={[34 45;44 53]};
>> x=[x1 x2 x3];
>> [a,b,c,d]=mychi2(x,'m');
```

表 19.2　不同地区的原料质量检测结果

	甲 地 区	乙 地 区	丙 地 区
一级	52	60	50
二级	64	59	65
三级	24	52	74

解：

此类问题即为独立性检验，是 χ^2 检验的另一种检验方式，它用来检验两个变量间是否存在联系的问题。

此题检验问题的原假设为

$$H_0 : p_{ij} = p_{i\cdot} p_{\cdot j},\quad \forall i, j, 1 \leqslant i \leqslant r, 1 \leqslant i \leqslant s,$$

式中：r 和 s 分别为变量 X 与 Y 的等级。即如果变量 X 与 Y 是独立的，或者说是没有关系，则 X 和 Y 的联合概率应该等于 X 和 Y 的边缘概率之积。

则根据相应统计的原理，可编程计算如下。

```
>> x=[52 60 50;64 59 65;24 52 74];
>> [p,h,para]=mychi2(x,'s');
>> p=5.4135e-004          %概率值
>> b=1                    %拒绝原假设，即认为地区和原料等级间存在依赖关系
>>para=                   %从各相关系数可看出，产地与原料等级间的相关程度不同
    ph: 0.1991            %ψ 相关系数
     C: 0.1953            %列联相关系数
     V: 0.1408            %V 相关系数
```

例 4.9　某公司在一项议案的调查中，得到如表 19.3 所示的列联表。问哪些因素与态度有关？哪些因素与态度无关？

表 19.3　调查数据

态度（Z）	支 持		反 对		弃 权	
工种（Y）	蓝领	白领	蓝领	白领	蓝领	白领
性别（X）						
男	60	50	95	40	34	44
女	80	45	105	41	44	53

解：

此例题涉及对数线性模型。根据对数线性模型编程计算如下。

```
>> x1={[60 50;80 45]};x2={[95 40;105 41]};x3={[34 45;44 53]};
>> x=[x1 x2 x3];
>> [a,b,c,d]=mychi2(x,'m');
```

表 19.2　不同地区的原料质量检测结果

	甲 地 区	乙 地 区	丙 地 区
一级	52	60	50
二级	64	59	65
三级	24	52	74

解：

此类问题即为独立性检验，是 χ^2 检验的另一种检验方式，它用来检验两个变量间是否存在联系的问题。

此题检验问题的原假设为

$$H_0 : p_{ij} = p_{i\cdot}p_{\cdot j}, \quad \forall i, j, 1 \leqslant i \leqslant r, 1 \leqslant i \leqslant s,$$

式中：r 和 s 分别为变量 X 与 Y 的等级。即如果变量 X 与 Y 是独立的，或者说是没有关系，则 X 和 Y 的联合概率应该等于 X 和 Y 的边缘概率之积。

则根据相应统计的原理，可编程计算如下。

```
>> x=[52 60 50;64 59 65;24 52 74];
>> [p,h,para]=mychi2(x,'s');
>> p=5.4135e-004          %概率值
>> b=1                    %拒绝原假设，即认为地区和原料等级间存在依赖关系
>>para=                   %从各相关系数可看出，产地与原料等级间的相关程度不同
    ph: 0.1991            %ψ 相关系数
     C: 0.1953            %列联相关系数
     V: 0.1408            %V 相关系数
```

例 4.9　某公司在一项议案的调查中，得到如表 19.3 所示的列联表。问哪些因素与态度有关？哪些因素与态度无关？

表 19.3　调查数据

态度（Z）	支　持		反　对		弃　权	
工种（Y）	蓝领	白领	蓝领	白领	蓝领	白领
性别（X）						
男	60	50	95	40	34	44
女	80	45	105	41	44	53

解：

此例题涉及对数线性模型。根据对数线性模型编程计算如下。

```
>> x1={[60 50;80 45]};x2={[95 40;105 41]};x3={[34 45;44 53]};
>> x=[x1 x2 x3];
>> [a,b,c,d]=mychi2(x,'m');
```

计算结果如表 19.4 所示：

表 19.4 计算结果

模 型	自由度（df）	Pearsonχ^2	LRT L^2	h	结 论
（X，Y，Z）	7	35.4077	35.5664	1	X，Y，Z 不独立
（X， YZ）	5	2.7813	2.7826	0	X 与 Y，Z 独立
（Y， XZ）	5	35.0923	35.1330	1	Y 与 X，Z 不独立
（Z， XY）	6	34.6628	34.4739	1	Z 与 X，Y 不独立
（XY， XZ）	4	34.0410	34.0406	1	给定 X，Y 与 Z 不独立
（XY， YZ）	4	1.6893	1.6901	0	给定 Y，X 与 Z 独立
（XZ， YZ）	4	2.3488	2.3492	0	给定 Z，X 与 Y 独立

可以看出，无法抗拒性别 X 与其他两个因素的独立性，这表明性别与工种以及性别与态度之间都没有相关性，然而，变量 Y 与 Z 存在交互作用，也即不同工种的职工对提案的态度是不同的。

例 4.10 在属性选择过程中，秩检验是一种常用的方法。秩统计量是完全由样本的秩决定的统计量，秩的检验在非参数估计中占有极其重要的地位，其原因一是秩检验使用灵活，易于在各种检验问题中从直观出发构造出统计量来；二是线性秩统计量有完备的大样本理论，其在原假设下往往与分布无关；三是秩检验的使用，相对于其他方法而言，计算上不是很复杂；四是与其他常用的检验方法相比，性能并不差。

下面试用 Kruskal-Wallis 检验法分析 Iris 数据。

解：

此检验的统计量为

$$H=\frac{12}{n(n+1)}\sum_{i=1}^{k}\frac{R_i^2}{n_i}-3(n+1)$$

式中：R_i、n_i 为各组的秩和及样本个数，如果有相同的秩，则采用平均秩；n 为样本的总数。如果存在结（即秩相等），则进行修正

$$H_c=\frac{H}{1-\frac{1}{n^3-n}\sum_{l=1}^{g}(\tau_l^3-\tau)}$$

式中：g 为结的个数；τ_l 为相应结的长度（即样本的个数）。

当组数 k=3，且每组例数 $n_i\leqslant 5$，可查 H 界值表得到 p 值；如果组数大于 3，则 H 近似地服从于自由度为 k-1 的 χ^2 分布，可查 χ^2 界限值得到 p 值。

当 $p<0.05$ 时，可以拒绝原假设，即各组的数据分布不完全相同，从而可推断 $\boldsymbol{X}_i$ 变量与目标分类变量 $\boldsymbol{Y}$ 具有相关性。

根据以上原理，编程计算如下。

```
>> a=dlmread('D:\数据.txt');
>> x=a(:,1:4);
```

```
>>class=[ones(50,1);2*ones(50,1);3*ones(50,1)];
>> [p,h,d,z]=myrank(x,class);
>> p=1.1102e-016
>> h=1
>> d=4.8692    1.0000    2.0000         %两两总体间差异的检验，说明组间的差异都相对较大
     8.5785    1.0000    3.0000
     3.7093    2.0000    3.0000
>> z=2.3940                             %标准正态分布的分位数
```

例 4.11　在现实的数据集中，往往不可避免地会存在属性值缺失的情况，而且可能发生在生活的各个研究领域。虽然缺失数据的存在加强了系统表现的不确定性，使得这种不确定性更加难以把握，但包含缺失的属性值或者不完整数据集仍然包含某些重要的信息量，所以一般在数据挖掘前，对缺失数据的处理显得非常重要和必要。

可以有多种方法对缺失的数据集进行处理，基于朴素贝叶斯分类方法就是其中的一种。表 19.5 为一个简单的样本数据库。数据库中含有 15 条记录和 4 个属性：分别是 Income、Age、Gender、HomeOwner，其中缺失属性用“？”表示。试预测第 15 条记录中的缺失数据。

表 19.5　样本数据库

No.	Income	Age	Gender	HomeOwner
1	low	<30	female	no
2	low	<30	male	no
3	low	30−55	female	yes
4	low	30−55	female	no
5	low	>55	female	no
6	high	<30	male	yes
7	high	30−55	female	yes
8	high	30−55	male	yes
9	high	30−55	male	yes
10	high	30−55	male	no
11	high	>55	male	yes
12	?	30−55	female	yes
13	?	30−55	male	yes
14	?	<30	female	?
15	?	?	male	no

解：

在应用贝叶斯方法进行计算时，首先要确定属性间是否有关联。如果属性间相互独立，则可以用朴素贝叶斯方法进行计算；如果属性间有关联，则计算条件概率与朴素贝叶斯中有所不同。为了更加合理地求出缺失值，既要考虑到属性间的独立性，又考虑到关联性，此时可采用双尺度

贝叶斯公式。

给定一个数据集，有 N 条记录和 M 个属性 $X_1,X_2,\cdots,X_M$，$c_1,c_2,\cdots,c_L$ 是某 X_i 样本空间的划分，对样本空间的任一的事件 X，都有

$$W=\sum_{r=1}^{L}p(c_r)p(X\mid c_r)$$

$$p_1=\frac{p(c_k)\prod_{i=1}^{M-1}p(x_i\mid c_k)}{W}\quad k=1,2,\cdots,L$$

$$p_2=\frac{p(c_k)\min(p(x_i\mid c_k))}{W}$$

$$\theta=p(X\mid c_k)$$

$$p(c_k\mid X)'=p_1+(p_2-p_1)\times\theta$$

W 即为全概公式，可以不计算出结果；p_1 为朴素贝叶斯方法的计算公式，p_2 为一般贝叶斯公式的改进形式；θ 为偏向独立或者关联的修正因子，其值越小，说明属性间独立性越大；反之是关联性强。

据此可以编程计算，从计算结果可看出，当某一行样本值中属性值缺失较多，则情况较为复杂，需考虑多种情况。

```
>>data ={'low' '<30'  'female'  'no';'low' '<30' 'male' 'no';'low' '30-55' 'female 'yes'
        'low' '30-55' 'female' 'no';'low' '>55' 'female' 'no';'high' '<30 'male' 'yes'
        'high' '30-55''female' 'yes';'high''30-55''male''yes';'high' '30-55' 'male' 'yes'
        'high' '30-55' 'male'  'no'; 'high' '>55'  'male' 'yes'; '? '  '30-55' 'female' 'yes';
        '?'   '30-55''male'  'yes';'?'   '<30' 'female' '?';'? ' '?'    'male'  'no'};
>> y=datafill(data,'?');
>> y{1}=name: {'? '  '30-55'  'female'  'yes'}
      val: [0.5800 0.4200]
      pro: {'high'}                  %最大可能的取值
>> y{2}=name: {'?'  '30-55'  'male'  'yes'}
       val: [0.9755 0.0245]
       pro: {'high'}
>> y{3}{1}= name: {'?'  '<30'  'female'  'no'}
         val: [0.0191 0.9809]
         pro: {'low'}
>> y{3}{2}=name: {'?'  '<30'  'female'  'yes'}
         val: [0.3027 0.6973]
         pro: {'low'}            %最大可能的取值
```

例 4.12　在数据预处理过程中，为了消除随机误差和噪声，经常会用到数据滤波及数据平滑

等技术。滤波技术中最简单的是移动均值滤波，通过在预先设定的窗口内，取所有数据的权重线性均值，即得滤波数据。窗口的大小定义为滤波宽度，随着窗口的移动，即得经滤波后的一系列数据。

在移动窗口均值滤波中，所有原始数据均给以相同的权重，这样往往使数据扭曲，若给数据以不同的权重，则可获得更有效的数据平滑。Savitzky-golay 就是这样一种滤波技术，它利用高次多项式来进行数据平滑，也称为卷积平滑。它能够保留原始数据中有用信息，是消除随机噪声的有效的平滑方法。

试对以下数据进行平滑处理。

```
x =[0.1580 0.2400 0.3750 0.4600 0.5860 0.6750 0.7200 0.7360 0.6700 0.5850 0.4550 0.3130 0.2140 0.1100 0.0670 0.0370]
```

解：

根据数据平滑的原理，可编程计算如下，

```
>>x=[0.158 0.240 0.375 0.460 0.586 0.675 0.720 0.736 0.670 0.585 0.455 0.313 0.214 0.110 0.067 0.037];
>>y=moving(x,7,4);          %七点四次平滑，得图 19.11
```

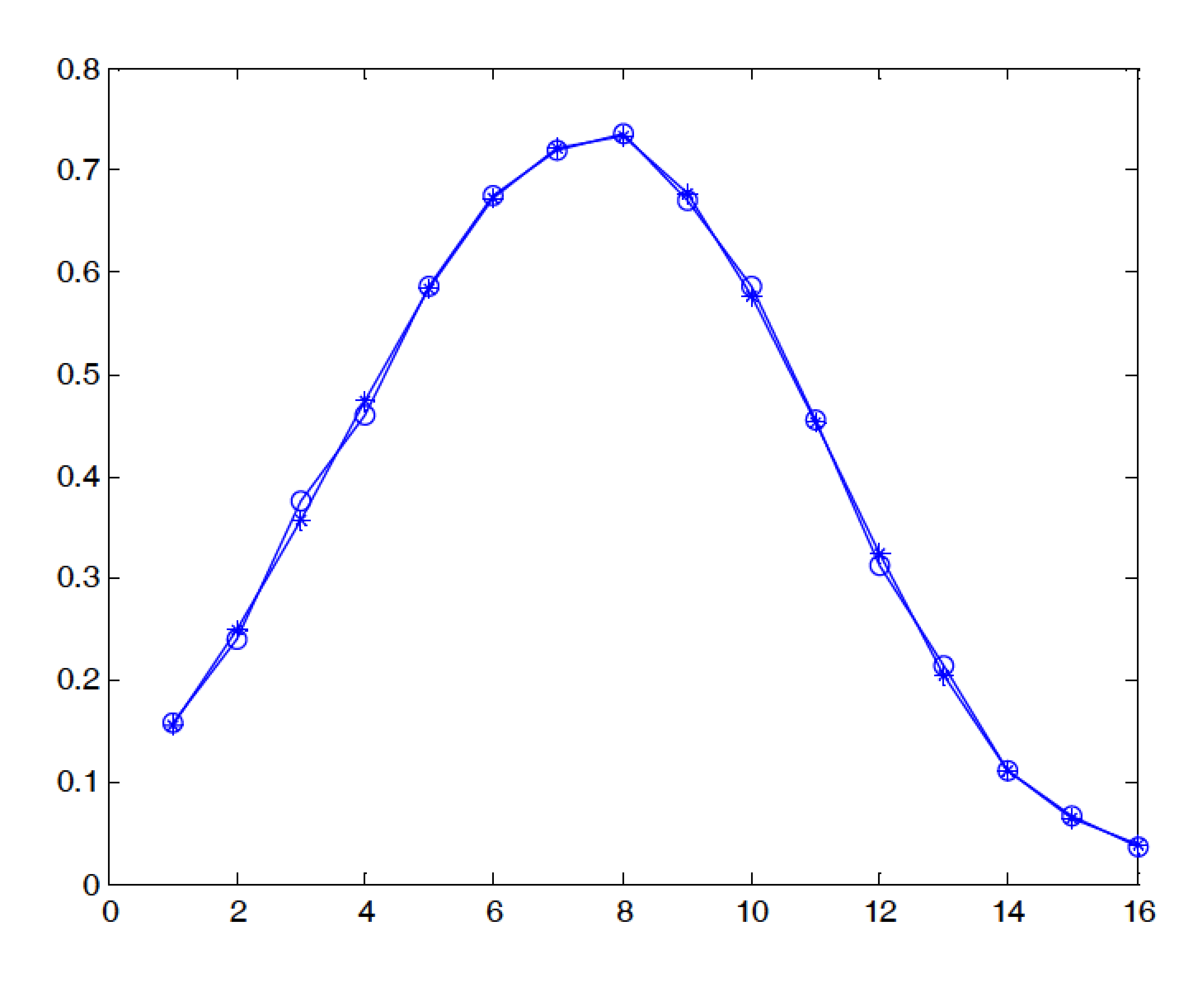


图 19.11　原始数据及平滑数据曲线

从图中可看出，原始数据及平滑后的数据曲线几乎重叠。

例 4.13　在数据挖掘的实际应用中，经常会遇到变量缺失问题。这会导致一些不能处理缺失值的分析方法无法应用。

当在处理含有缺失值的数据时，可以运用以下几种最常见的策略。

（1）将含有缺失值的样本删除。

（2）根据变量间的相关关系填补缺失值；

（3）根据样本间的相似性填补缺失值。

（4）使用能够处理缺失值数据的工具。

试对给出的数据集进行缺失数据的填补。

解：

```
>>a1=importdata('D:\data.txt');
>>a1=1.0e+004*a1;
>>y=filldata(a1);
```

按照提示进行缺失值的填补。需要说明的是，在函数中缺失值用 NaN 代替，也可以用其他符号来表示。

第 20 章

分类

20.1 分类概述

分类就是利用训练数据中学习到的规律来确定未知样本的类别。例如根据银行客户信用贷款的历史数据，使用分类可以构造“拖欠贷款”和“非拖欠贷款”两类客户的模型，对于将要申请信用贷款的客户，可以根据分类模型和该客户的特征来预测该客户是否会拖欠贷款，从而决定是否同意给该客户贷款。

分类是数据挖掘中一项非常重要的任务，在各个领域得到了广泛的应用，如图像与模式识别、医疗诊断、故障诊断以及金融市场走势分类等。在银行、保险等领域中，可以利用已有数据建立分类模型，评估客户的作用等级；在市场营销中，可以利用历史数据的销售数据，预测某些商品是否可以销售，预测广告应该投放到哪些区域，及预测某客户是否会成为商场客户从而实施定点传单投放等。

20.2 方法

数据分类可以分为三个步骤：①将数据集划分为两部分，一部分作为训练集，另一部分作为测试集；②通过分析训练集的特点来构造分类模型；③对测试集建立的分类模型进行分类，评估该分类模型的分类准确度，通常使用分类准确度高的分类模型对类标号未知的样本数据进行分类。

为建立模型而被分析的数据集称为训练集，其中单个元组称为训练样本。每个训练样本包括多个属性，其中有一个属性决定该元组属于一个预定义的类，该属性称为类标号属性或目标属性，其他属性称为预测属性。预测属性按性质分，可以分为类别属性和数值属性。分类的目的就是对训练样本进行分析，根据其预测属性特征，得出一个精确的分类模型，据此对目标属性未知的元组进行类归属判断。

分类器的构造方法有基于统计的方法、基于距离的算法、基于决策树的算法、基于神经网络的算法、基于规则的算法及组合技术。

基于统计的算法包括回归法、贝叶斯分类等；基于距离（即相似度）的算法有 K-最近邻法等；基于决策树的算法有 ID3、C4.5 和 C5.0、CART 等；基于神经网络的算法主要是 BP 算法。除此外，还有其他的分类算法，如粗糙集、支持向量机算法等。

不同的分类器有不同的特点，分类器的评价或比较尺度有以下一些关键性指标。

（1）分类准确率。指的是模型正确地预测新的或先前未见过的数据的类标号的能力。通常分类算法寻找的是分类准确率高的分类模型。影响分类准确率的因素有训练数据集质量、记录的数目、属性的数目、属性中的信息和测试数据集记录的分布等。

评估分类器准确率常见的方法有保持方法、留一法、自展法、K-折交验证等。保持方法将给定数据随机地划分成两个独立的集合，即训练集和测试集。通常将 2/3 的数据分配到训练集，其余 1/3 分配到测试集。首先使用训练集导出分类法，然后在测试集上评估准确度。随机子选样是保持方法的一种变形，它将保持方法重复 k 次，取每次迭代准确度的平均值作为总体精度估计。

留一法是在每一阶段留出一个数据点，但每个数据点是依次留出的，所以测试集的大小等于整个训练集的大小。每个仅含一个数据点的测试集独立于它所测试的模型。

自展法是利用样本和从样本中轮番抽出的同样容量的子样本间的关系，对未知的真实分布和

样本的关系建模。

在 k-折交验证法中，原始数据被划分成 k 个互不相交的子集或“折” $S_1,S_2,\cdots,S_k$，每个折的大小大致相等，进行 k 次训练和测试，在第 i 次迭代时，S_i 用作测试集，其余的子集都用于训练分类法。分类准确度估计是 k 次迭代正确分类数据除以初始数据的样本总数。在分层交叉验证中，将每个折分层，使得每个折中样本的类分布与初始数据中的大致相同。

另外还应注意分类的效果一般和数据特点有关，有的数据噪声较大，有的有缺失值，有的分布稀疏，有的字段或属性间相关性强，有的属性是离散的而有的是连续值或混合式的。

（2）计算复杂度。计算复杂度决定着算法执行的和占用的资源，它依赖于具体的实施细节和软硬件环境。由于数据挖掘中的操作对象是海量的数据库，因而空间和时间的复杂度将是非常重要的问题。

（3）可解释性。分类结果只有可解释性好及容易理解，才能更好地用于决策支持。结果的可解释性越好，算法受欢迎的程度越高。

（4）可扩展性。可扩展性是指在给定内存和磁盘空间等可用的系统资源的前提下，算法的运行时间应当随数据库大小线性增加。

（5）鲁棒性。它是指在数据集中含有噪声和缺失值的情况下，仍具有较好的正确分类数据的能力。

（6）累积增益图。累积增益图会在给定的类别中显示通过把个案总数的成分比作为目标“增益”的个案总数的百分比。对角线是“基线”曲线，曲线离基线的上方越远，增益越大。累积增益图通过选择对应的大量增益的百分比选择分类标准值，然后将百分比与适当分界值映射。

（7）不平衡数据分类。不平衡分类是指训练样本数量在类间分布不平衡。具体地说就是在同一数据集中某些类的样本数远大于其他类的样本数，其中样本少的类为少数类（正类），样本多的类为多数类（负类）。具有不平衡分布的数据集出现在许多实际应用中，很多重要信息隐藏在少数类中。在不平衡数据分类中，少数类的正确分类比多数类的正确分类更有价值，仅用准确率评价分类模型并不合适。

对于不平衡数据集的分类，常用度量除分类准确率外，还有精度、召回率和 F_1 度量。

- 精度定义为正确分类的正例（TP）个数占分类为正例的样本个数的比例

$$p=\frac{TP}{TP+FP}$$

- 召回率定义为正确分类的正例个数占实际正例个数的比例

$$r=\frac{\text{被正确分类的正例样本个数}}{\text{实际正例样本个数}}=\frac{TP}{TP+FN}$$

- F_1 度量表示精度和召回率的调和平均值：$F_1=\dfrac{2rp}{r+p}$，F_1 度量趋向于接近精度和召回率的较小者。

许多数据挖掘方法在不平衡数据集上的性能不佳，因为少数类中的规律会被多数类中的规律所掩盖。一般的分类方法认为所有的错误分类代价是相同的，但在实际应用中往往不同类的错分代价是不同的。针对不平衡数据的分类主要有两种策略：一是通过过抽样和欠抽样改变不同类别的记录比例（过抽样是通过对含正类的元组重复采样来增加含正类的个数；欠抽样则是通过随机

地删除含负类的元组来减少负类的个数），减少类别不平衡的程度；二是引入代价敏感机制，通过代价最小化来分类数据。

对于一个给定的分类问题，没有一种分类技术总是产生最好的结果，每种技术都各有优缺点。因此可以采用使用组合技术来提高分类精度，也即将多个分类学习方法聚集在一起来提高分类准确度和模型的稳定性。

组合分类方法并不是简单地将数据集在多个不同分类器上重复训练，而是对数据集进行扰动，另外，一个分类器训练中的错误还可以被下一个分类器利用。通过扰动，分类器能学习到更一般的模型，从而消除单个分类器所产生的偏差，得到更为精确的模型。

组合分类技术从组合的内容来源看，组合分类器可以分成两大类：一类是将不同种类的分类器进行组合，通过组合弥补各种分类器间的不足。该类分类方法将各种法则通过某种算法组合起来，以得出各个分类法的优点，从而达到改善和提高分类精度的目的。另一类则是将原始数据分为若干维度，对不同维度用相同分类器进行分类处理，最后分类结果进行组合表决获得总的分类结果。从数学上看，其本质是完成高维空间的低维计算并做成非线性合成。

就组合结构而言，组合分类器可分为级联和并联两种形式。其中级联结构是将单分类器的输出作为另一个单分类器的输入。

并联结构则各个分类器的输出是相互独立的，最后再利用某种方法将相互独立的分类输出信息组合起来，作为最后组合分类器的输出。一般而言，并联结构更具有现实意义。在该方式下各个单分类器的设计完全独立，不必考虑其他分类器输出信息的影响，有利于将各自独立的子分类器组合成一个高效能的分类识别系统。

在实际应用中，组合分类器可以有多种多样的设计。例如对数据集抽取的训练集，可以是随机提取一个，作为所有基分类器的训练集，也可以分别从数据集中有放回地随机抽取样本容量相同（或不同）、但数据元组（个体）不同的训练集，分别来训练各基分类器。还可以对数据集中的个体，赋予不同权重，使个体被抽到训练集中的机会不一样。另外还可以将相同的表决权赋予每个基分类器，也可以对它们赋予不同的权重，此权重大小可根据各基分类器的准确率确定，分类器准确率越高，它的表决权重就越高。

总之，建立组合分类器时，其基本思想是，使各分类器能够互补，能更好地降低噪声数据和过拟合的影响，使组合分类器的准确率显著高于各基分类器。

组合分类技术克服了单一分类器的诸多缺点，如对样本的敏感性，难以提高分类精度等，但它必须满足基分类器之间的完全独立的条件，在实践上很难达到这个条件。虽然与独立分类器相比，组合分类器分类精度会有一定程度的提高，但提高的程度不大，甚至出现组合后分类精度降低的情况。因此对于组合分类器而言，各子分类器的组合策略，即采用何种方式将各子分类器集成以达到各子分类器分类结果的有效互补，成为利用组合分类方法进行分类处理的关键。

事实上没有一种分类方法对于所有数据类型和实际领域都优于其他方法，所以选择一些合适的分类器，搭配组合所得的组合分类器，通常都能使其性能超过单个分类器。

多种分类技术已在相关章节做了介绍，在此不再重复，仅介绍各类算法在实际分类中的应用。

地删除含负类的元组来减少负类的个数），减少类别不平衡的程度；二是引入代价敏感机制，通过代价最小化来分类数据。

对于一个给定的分类问题，没有一种分类技术总是产生最好的结果，每种技术都各有优缺点。因此可以采用使用组合技术来提高分类精度，也即将多个分类学习方法聚集在一起来提高分类准确度和模型的稳定性。

组合分类方法并不是简单地将数据集在多个不同分类器上重复训练，而是对数据集进行扰动，另外，一个分类器训练中的错误还可以被下一个分类器利用。通过扰动，分类器能学习到更一般的模型，从而消除单个分类器所产生的偏差，得到更为精确的模型。

组合分类技术从组合的内容来源看，组合分类器可以分成两大类：一类是将不同种类的分类器进行组合，通过组合弥补各种分类器间的不足。该类分类方法将各种法则通过某种算法组合起来，以得出各个分类法的优点，从而达到改善和提高分类精度的目的。另一类则是将原始数据分为若干维度，对不同维度用相同分类器进行分类处理，最后分类结果进行组合表决获得总的分类结果。从数学上看，其本质是完成高维空间的低维计算并做成非线性合成。

就组合结构而言，组合分类器可分为级联和并联两种形式。其中级联结构是将单分类器的输出作为另一个单分类器的输入。

并联结构则各个分类器的输出是相互独立的，最后再利用某种方法将相互独立的分类输出信息组合起来，作为最后组合分类器的输出。一般而言，并联结构更具有现实意义。在该方式下各个单分类器的设计完全独立，不必考虑其他分类器输出信息的影响，有利于将各自独立的子分类器组合成一个高效能的分类识别系统。

在实际应用中，组合分类器可以有多种多样的设计。例如对数据集抽取的训练集，可以是随机提取一个，作为所有基分类器的训练集，也可以分别从数据集中有放回地随机抽取样本容量相同（或不同）、但数据元组（个体）不同的训练集，分别来训练各基分类器。还可以对数据集中的个体，赋予不同权重，使个体被抽到训练集中的机会不一样。另外还可以将相同的表决权赋予每个基分类器，也可以对它们赋予不同的权重，此权重大小可根据各基分类器的准确率确定，分类器准确率越高，它的表决权重就越高。

总之，建立组合分类器时，其基本思想是，使各分类器能够互补，能更好地降低噪声数据和过拟合的影响，使组合分类器的准确率显著高于各基分类器。

组合分类技术克服了单一分类器的诸多缺点，如对样本的敏感性，难以提高分类精度等，但它必须满足基分类器之间的完全独立的条件，在实践上很难达到这个条件。虽然与独立分类器相比，组合分类器分类精度会有一定程度的提高，但提高的程度不大，甚至出现组合后分类精度降低的情况。因此对于组合分类器而言，各子分类器的组合策略，即采用何种方式将各子分类器集成以达到各子分类器分类结果的有效互补，成为利用组合分类方法进行分类处理的关键。

事实上没有一种分类方法对于所有数据类型和实际领域都优于其他方法，所以选择一些合适的分类器，搭配组合所得的组合分类器，通常都能使其性能超过单个分类器。

多种分类技术已在相关章节做了介绍，在此不再重复，仅介绍各类算法在实际分类中的应用。

20.3 例题

例 4.14 为了解耕地的污染状况与水平，从 3 块由不同水质灌溉的农田里各取 6 个样品，每个样品均作土壤中铜、镉、氟、锌、汞和硫化物等 7 个变量的浓度分析，原始数据如表 20.1 所示。试确定 3 个待判样品所属组别。

表 20.1 原始数据 单位：mg/kg

	序 号	x_1	x_2	x_3	x_4
第一组	1	11.853	0.480	14.360	25.210
	2	45.596	0.526	13.850	24.040
	3	3.525	0.086	24.400	49.300
	4	3.681	0.327	13.570	25.120
	5	48.287	0.386	14.500	25.900
第二组	1	4.741	0.140	6.900	15.700
	2	4.223	0.340	3.800	7.100
	3	6.442	0.190	4.700	9.100
	4	16.234	0.390	3.400	5.400
	5	10.585	0.420	2.400	4.700
第三组	1	48.621	0.082	2.057	3.847
	2	288.149	0.148	1.763	2.968
	3	316.604	0.317	1.453	2.432
	4	307.310	0.173	1.627	2.729
	5	82.170	0.105	1.217	2.188
待判样	1	3.777	0.870	15.400	28.200
	2	62.856	0.340	5.200	9.000
	3	3.299	0.180	3.000	5.200

解：

可以用多种方法对数据进行分类。

1. 辅助方法

利用各种可视化方法对数据进行显示，从中可粗略地分析出各样品在空间的分布情况。

```
>> data1=guiyi(data,1);
>>y=pdist(data1);z=linkage(y,'single');h=dendrogram(z);    %图 20.1，冰柱图
>>star(data,2);                                            %图 20.2
>>y=myNLM(data,40);                                        %图 20.3
```

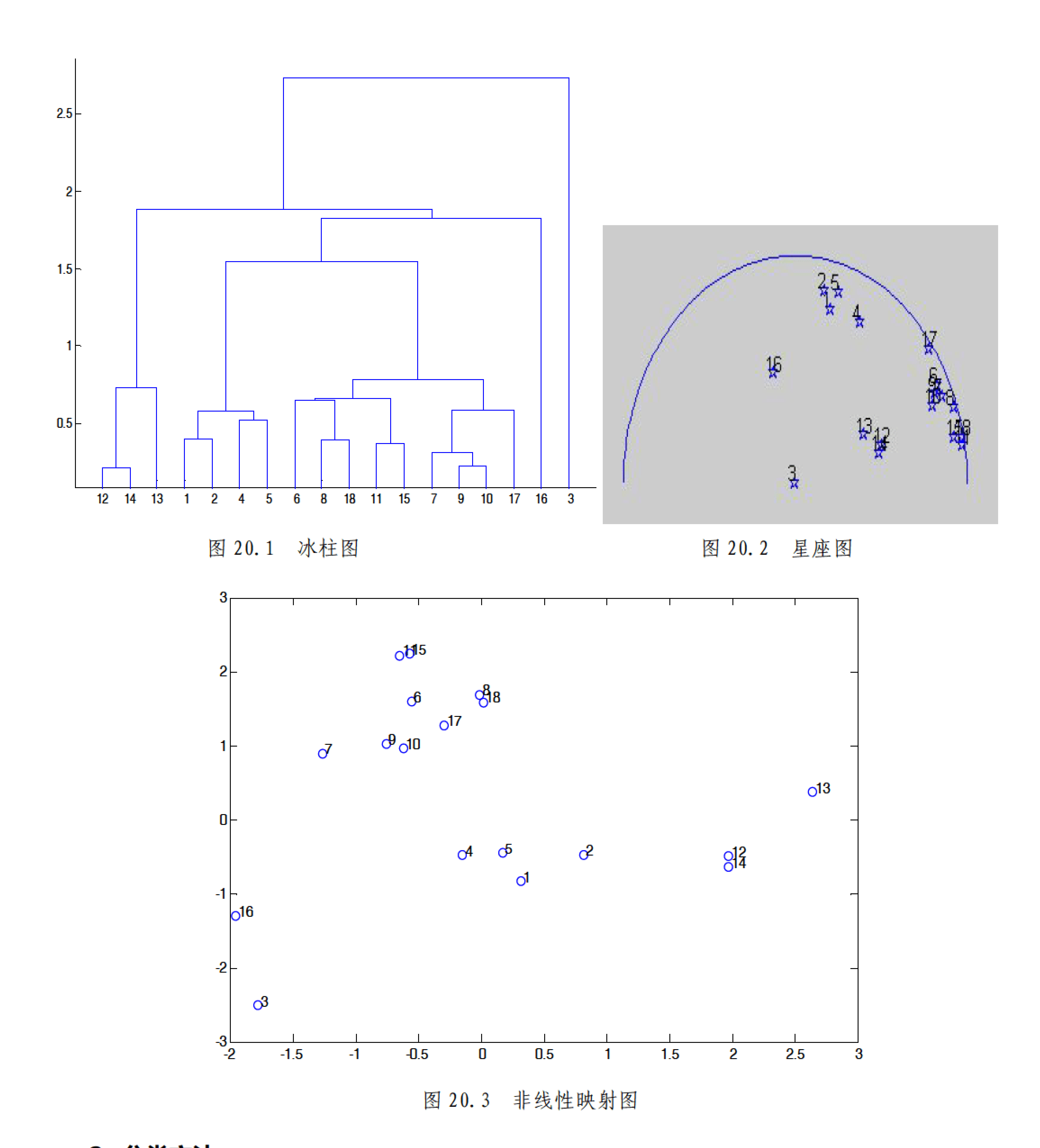


图 20.1　冰柱图

图 20.2　星座图

图 20.3　非线性映射图

2. 分类方法

```
>>sample=[3.7770 0.8700 15.4000 28.2000;62.8560 0.3400 5.2000 9.0000;
          3.2990 0.1800 3.0000 5.2000];
>> y=classify(sample,class,[1; 1; 1; 1; 1; 2 ;2 ;2; 2; 2 ;3; 3; 3 ;3 ;3])
                                                    %matlab 自有函数
  y=1   2   2                                       %分类结果
>>y=knnclassify(sample,class,[1; 1; 1; 1; 1; 2 ;2 ;2; 2; 2 ;3; 3; 3 ;3 ;3])
```

```
                                                      %k－近邻法函数
  y =1  3  2
>> T=treefit(class,[1; 1; 1; 1; 1; 2 ;2 ;2; 2; 2 ;3; 3; 3 ;3 ;3]);%决策树函数
>> y=treeval(T,sample)
   y=1  3  2
>>result=fisher(class1,class2,class3,sample);
   result =1  2   2                          %fisher 分类法
 >>result=bayes(class1,class2,class3,sample,1)
   result=2  1   2                           %基于最小错误率的 bayes 分类法
>> y=BPclass({class1;class2;class3},sample);  %BP 神经网络分类
  y=1.6145    1.3820    3.0583               %分类结果，即 2  1  3
```

例 4.15 决策树算法是一种逼近离散函数值的方法。它是一种典型的分类方法，首先对数据进行处理，利用归纳算法生成可读的规则和决策树，然后使用决策对新数据进行分析。本质上决策树是通过一系列规则对数据进行分类的过程。它具有分类精度高、生成的模式简单、对噪声数据有很好的健壮性等优点，是目前应用最为广泛的归纳推理算法之一，在数据挖掘中受到研究者的广泛关注。

MATLAB 中自带有决策树算法函数（classsregtree），现利用此函数对 matlab 自有的 fisheriris 数据进行分类分析。

解：

```
>> a=load('fisheriris');
>> t=classregtree(meas, species,'names',{'SL' 'SW' 'PL' 'PW'}); %决策树分类
>> view(t);                                          %显示决策树，图 20.4
```

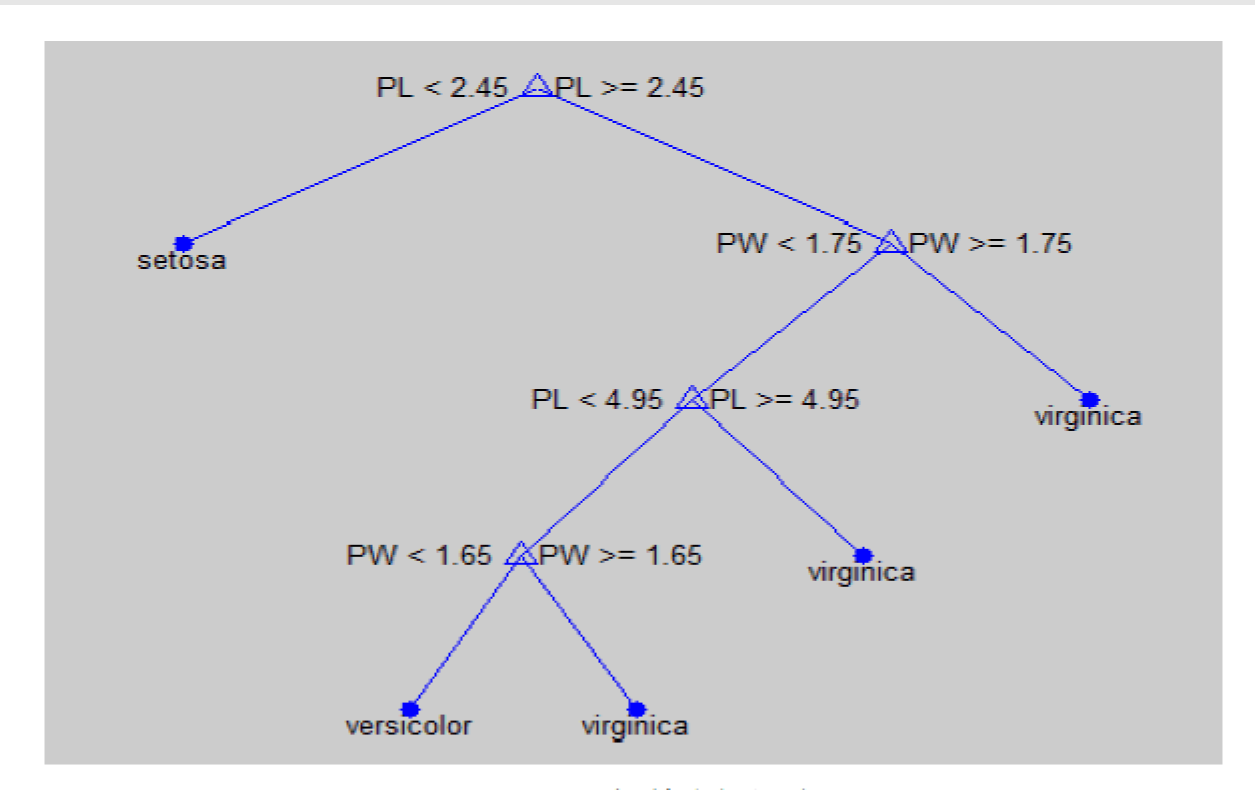


图 20.4　决策树分类图

例 4.16 在分类（判别）分析中，如果样品集为高维数据集，则建立判别函数需要大量的计算时间，而且由于有关矩阵的阶数太高，使解的精度下降，甚至由于变量的不独立性而引起计算上的困难。另一方面，由于不太重要的变量的引入，产生干扰而影响判别效果，有时还会增加错误的次数。因此，在可供判别的自变量中选出显著性变量是很重要的。

变量的选择（即降维）除了可采用粗糙集、主成分分析等方法外，还可以进行逐步判别分析，

其原理与逐步回归的基本思想相似，即都采用“有进有出”的算法，即每一步都进行检验，把一个“最重要”的变量选入判别式，同时也考虑较早进入判别式的某些变量，如果其“重要性”也随着其后一些变量的选入而变化，已失去原有的重要性时（被某些变量的作用所代替），应把它及时地从判别式中剔除出去，使最终的差别式仅仅保留“重要”的变量。

从经验得知，可以有病人心电图的 5 个指标来区分健康人（类 1），主动脉硬化症患者（类 2）及冠心病患者（类 3）三类人。其经验数据如表 20.2 所示。试找出判别函数，一个病人的心电图中，x_1=7.40、x_2=267.88、x_3=14.40、x_4=5.70、x_5=10.66，该病人应归入哪一类?

解：

根据逐步判别的原理，可编程计算如下。

```
>>load mydata;
>>train={x1,x2,x3};sample=[7.40 267.88 14.40 5.70 10.66];
>> [a,b,c]=stepclass(train,sample); %逐步判别函数
>> a=2  5                          %判别函数中的变量号，即第2个及第5个属性用于分类
   b= q: [-0.7376 -1.1896 -1.5261]  %判别函数中的系数
     c0: [-9.2963 -16.4467 -16.7832]
     c1: [3×2 double]
   c=3                             %样品的分类结果
```

表 20.2　经验数据

类　别	x_1	x_2	x_3	x_4	x_5
类 1	8.11	261.01	13.23	5.46	7.36
	9.36	185.39	9.02	5.66	5.99
	9.85	249.58	15.61	6.06	6.11
	2.55	137.13	9.21	6.11	4.35
	6.01	231.34	14.27	5.21	8.79
	9.64	231.38	13.03	4.88	8.53
	4.11	260.25	14.72	5.36	10.02
	8.90	259.51	14.16	4.91	9.79
	7.71	273.84	16.01	5.15	8.79
	7.51	303.59	19.14	5.70	8.53
	8.06	231.03	14.41	5.72	6.15
类 2	6.80	308.90	15.11	5.52	8.49
	8.68	258.69	14.02	4.79	7.16
	5.67	355.54	15.13	4.97	9.43
	8.10	476.69	17.38	5.32	11.32
	3.71	316.12	17.12	6.04	8.17
	5.37	274.57	16.75	4.98	9.67
	9.89	409.42	19.47	5.19	10.49

续表

类　别	x_1	x_2	x_3	x_4	x_5
类 3	5.22	330.34	18.19	4.96	9.61
	4.71	331.47	21.16	4.30	13.72
	4.71	352.50	20.79	5.07	11.00
	3.26	347.31	17.90	4.65	11.19
	8.27	189.56	12.74	5.46	6.94
测试样本	7.40	267.88	14.40	5.70	10.66

例 4.17　集成学习法是将多个分类方法聚集在一起来提高分类准确率。通常一个集成分类器的分类性能会好于单个分类器。集成学习法由训练数据构建一组基分类器，然后通过对每个基分类器的预测进行投票来进行分类。

构建集成分类器的方法主要包括装袋（bagging）、提升（boosting）、AdaBoost 算法等。其中 AdaBoost 算法是由 Yoav Freund 和 Robert Schapire 提出的最重要的集成学习算法，该算法具有可靠的理论基础、精确的分类精度、简单等优点。

利用该算法对 ionosphere 数据集进行分类。

解：

AdaBoost 算法的伪代码如下。

函数：AdaBoost($\boldsymbol{D}$,T)

输入：样本数据集 $\boldsymbol{D}$，学习提升轮数 T

输出：集成分类器 $H(x)$

（1）初始化 N 个样本的权重 $\boldsymbol{W}_1(x)=1/N\ (i=1,2,\cdots,N)$

（2）for t=1 to T do

（3）根据权重 $\boldsymbol{W}_t$的分布，通过对 $\boldsymbol{D}$ 进行有放回抽样产生训练集 $\boldsymbol{D}_t$

（4）在 $\boldsymbol{D}_t$上训练产生一个弱学习器（基分类器）h_t

（5）用 h_t对原训练集 D 中所有样本进行分类，并度量 h_t的误差

$$e_t=\frac{1}{N}\left[\sum_{i=1}^{N}W_t(i)I(h_t(x_i)\neq y_i)\right]$$

（如果 $(h_t(x_i)\neq y_i$ 为真，则 $I(h_t(x_i)\neq y_i)=1$，否则为 0）

（6）if e_t>0.5 then

（7）重新将权重初始化为 $1/N$，转步骤（3）重试

（8）end if

（9）决定 h_t的权重

$$\alpha_t=\frac{1}{N}\ln\left(\frac{1-e_t}{e_t}\right)$$

（10）更新权重分布

$$W_{t+1}(i)=\frac{W_t(i)}{Z_t}\times\begin{cases}\exp(-\alpha_t) & if\ \ h_t(x_i)=y_i\\ \exp(\alpha_t) & if\ \ h_t(x_i)\neq y_i\end{cases}=\frac{W_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

其中：Z_t是一个正规因子，用来确保$\sum_t W_{t+1}=1$。

（11）end for

（12）$H(x)=\text{sign}(\sum_{t=1}^{T}\alpha_t h_t(x))$

其中：sign()为符号函数，$\text{sign}(\sum_{t=1}^{T}\alpha_t h_t(x))=\begin{cases}1 & if\ \ \sum_{t=1}^{T}\alpha_t h_t(x)\geq 0\\ -1 & if\ \ \sum_{t=1}^{T}\alpha_t h_t(x)<0\end{cases}$

据此，便可以编程计算。在此利用从 mathworks 网站上下载的 AdaBoost 算法工具箱 GML_AdaBoost_Matlab_Toolbox_0.3 进行计算。

```
>> file_data = load('Ionosphere.txt');Data = file_data(:,1:end-1)';Labels =
file_data(:, end)';
>>Labels = Labels*2 - 1;                                    % 只能处理 1，-1 两类问题
>>MaxIter = 100;                                                 %提升轮数
>>TrainData= Data(:,1:2:end);TrainLabels = Labels(1:2:end); %将数据集分类
>>ControlData = Data(:,2:2:end);ControlLabels = Labels(2:2:end);
>>weak_learner= tree_node_w(3);                                  %构建弱分类器
```

以下为 adaBoost 算法，其中 Real AdaBoost 是一般 AdaBoost 算法（generalization of a basic AdaBoost），Modest AdaBoost 则是为了防止过拟合的更一般化 AdaBoost 算法（regularized tradeoff of AdaBoost）：

```
>> [RLearners RWeights]= RealAdaBoost(weak_learner, TrainData, TrainLabels,
MaxIter);
>>[MLearners MWeights]= ModestAdaBoost(weak_learner, TrainData, TrainLabels,
MaxIter);
>>ResultR=sign(Classify(RLearners, RWeights, ControlData));
>>ResultM=sign(Classify(MLearners, MWeights, ControlData));
>>ErrorR = sum(ControlLabels ~= ResultR) / length(ControlLabels)
>>ErrorM = sum(ControlLabels ~= ResultM) / length(ControlLabels)     %错误率
   ErrorR = 0.0629       ErrorM = 0.0686
```

第 21 章

预　测

21.1　回归分析

预测是构造和使用模型评估无标号样本类，或评估给定样本可能具有的属性值或区间值。预测的目的是从历史数据中自动推导出对给定数据的推广描述，从而能对未来数据进行预测。

预测技术一般采用回归统计方法，包括线性回归、非线性回归、多元回归等。

回归分析主要用于了解自变量与因变量的数量关系，主要用于寻找两个或两个以上的变量之间互相变化的关系，并借此了解变量间的相关性，可用以通过控制自变量来影响因变量，也可进一步通过回归分析来进行预测。利用数据库某些有用的信息，就可以对未知的变量进行预测。

在回归分析中，要注意在考虑自变量的选取时，必须要注意所选出的自变量与因变量是否存在因果关系。它们的选择，可以根据相关理论或逻辑或根据研究人员探讨的变量关系来决定。

回归分析的步骤如下（如图 21.1 所示）。

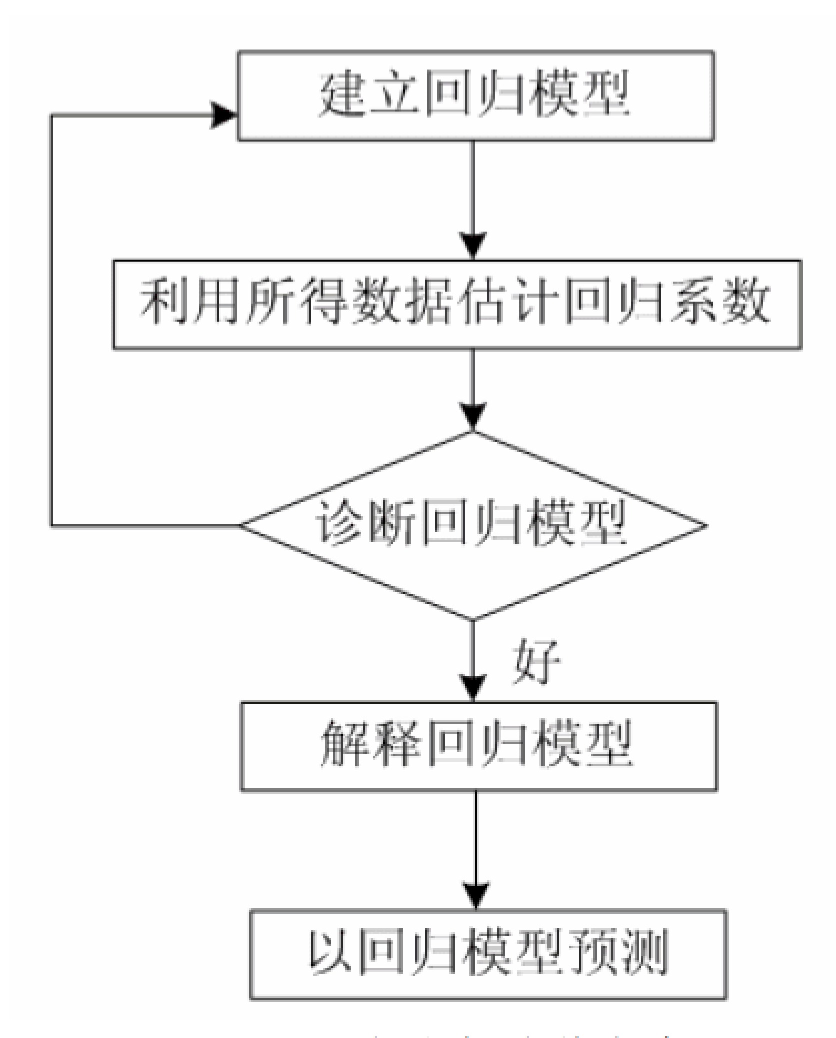


图 21.1　回归分析的基本步骤

（1）由分布图的情况或专门学科的知识，拟定测定值间的数学模型。

（2）用最小二乘法（或其他规则）尝试正规方程式。

（3）确定回归方程式。

（4）用图查看所求的方程曲线与测定值的分布是否一致，以确定所选的数学模型是否合理。

回归分析包括线性回归、非线性回归、多元回归、泊松回归、对数回归、主成分回归等。许多问题可以用线性回归解决，还有的问题可以通过对变量进行变换，将非线性问题转换成线性问题来处理。

回归分析的基本原理及方法已在第 2 篇“数据挖掘算法”做了介绍，在此主要介绍逐步回归、岭回归及主成分回归分析。

21.1.1　逐步回归

实际问题中影响因变量的因素可能很多，我们希望从中选择出影响显著的自变量来建立回归模型，这就涉及变量的选择问题。如果自变量选得太少，则自变量对 Y（因变量）的决定系数太

21.1 回归分析

预测是构造和使用模型评估无标号样本类，或评估给定样本可能具有的属性值或区间值。预测的目的是从历史数据中自动推导出对给定数据的推广描述，从而能对未来数据进行预测。

预测技术一般采用回归统计方法，包括线性回归、非线性回归、多元回归等。

回归分析主要用于了解自变量与因变量的数量关系，主要用于寻找两个或两个以上的变量之间互相变化的关系，并借此了解变量间的相关性，可用以通过控制自变量来影响因变量，也可进一步通过回归分析来进行预测。利用数据库某些有用的信息，就可以对未知的变量进行预测。

在回归分析中，要注意在考虑自变量的选取时，必须要注意所选出的自变量与因变量是否存在因果关系。它们的选择，可以根据相关理论或逻辑或根据研究人员探讨的变量关系来决定。

回归分析的步骤如下（如图 21.1 所示）。

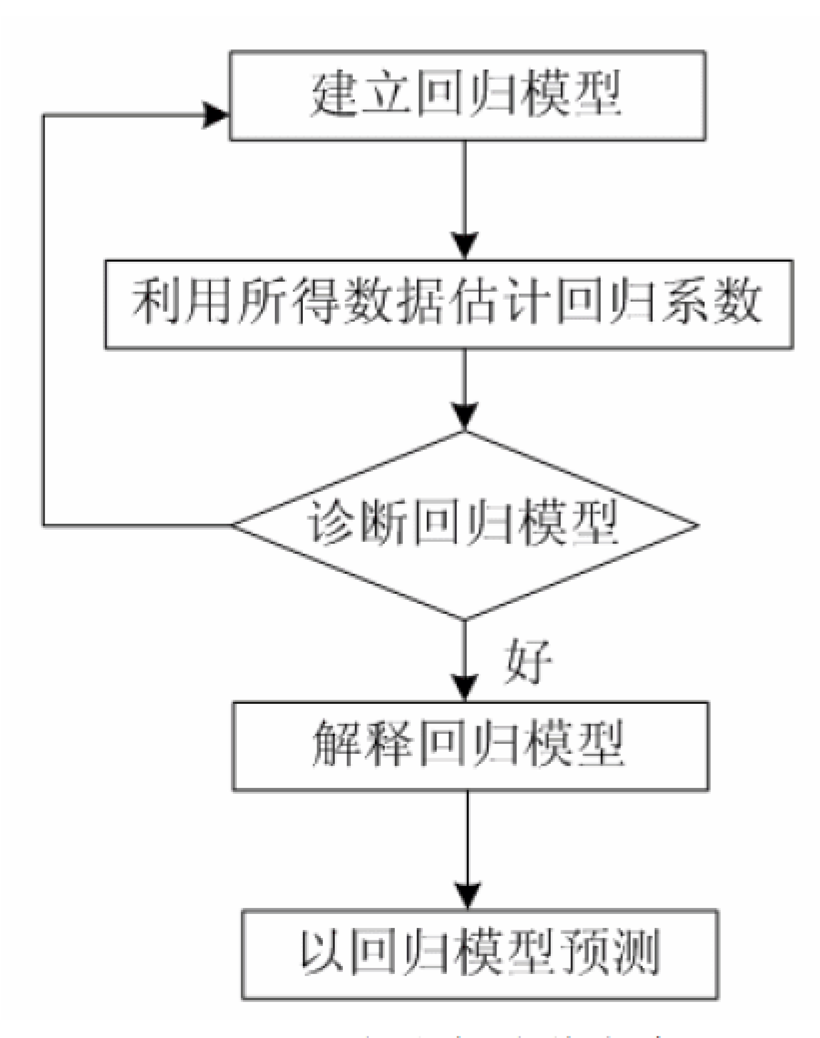


图 21.1 回归分析的基本步骤

（1）由分布图的情况或专门学科的知识，拟定测定值间的数学模型。

（2）用最小二乘法（或其他规则）尝试正规方程式。

（3）确定回归方程式。

（4）用图查看所求的方程曲线与测定值的分布是否一致，以确定所选的数学模型是否合理。

回归分析包括线性回归、非线性回归、多元回归、泊松回归、对数回归、主成分回归等。许多问题可以用线性回归解决，还有的问题可以通过对变量进行变换，将非线性问题转换成线性问题来处理。

回归分析的基本原理及方法已在第 2 篇“数据挖掘算法”做了介绍，在此主要介绍逐步回归、岭回归及主成分回归分析。

21.1.1 逐步回归

实际问题中影响因变量的因素可能很多，我们希望从中选择出影响显著的自变量来建立回归模型，这就涉及变量的选择问题。如果自变量选得太少，则自变量对 Y（因变量）的决定系数太

小，导致过大的偏差。但如果把与因变量有关的自变量都选中是不可能的。一般来讲，选的自变量越多，剩余平方和越小，然而多个自变量中有相当一部分对 Y 的影响不显著，反而会产生因自由度的减少而增大的误差；另外，多个自变量间的相关也会给回归方程的实际解释带来麻烦，即多重共线性的影响。基于以上原因，在作回归分析时一般要求进入回归方程的自变量都是显著的，未进入的自变量都是不显著的，即建立最优回归方程。

逐步回归法是建立最优回归方程的一种统计方法，其特点有两个：首先，对引入的因子进行检验，显著者引入，不显著者剔除；其次，每引入一个新因子，要对前面引入的因子进行检验，显著者保留，不显著者剔除，这样反复做下去，直至进入的因子都显著，未进入方程的因子都不显著为止，就得到了最优回归方程。

逐步回归中的基本思路为，先确定一初始子集，然后每次从子集外影响显著的变量中引入一个对 Y 影响最大的，再对原来子集中的变量进行检验，从变得不显著的变量中剔除一个影响最小的，直至不能引入和剔除为止。使用逐步回归有两点值得注意，一是要适当地选定引入变量的显著性水平 α_{in} 和剔除变量的显著性水平 α_{out}。显然，α_{in} 越大，引入的变量越多；α_{out} 越大，剔除的变量越少；二是由于各个变量的相关性，一个新的变量引入后，会使原来认为显著的某个变量变得不显著，从而被剔除，所以在最初选择变量时应尽量选择相互独立性强的自变量。

在具体操作中，要通过 F 检验才能得出变量的引入或剔除。

1. 引入标准

统计量：$F_i^{(l)} = \dfrac{V_i^{(l)}}{Q^{(l)}/(n-l-1)}$，服从 $F(l,n-l-1)$分布。

可以根据给出的置信度，从 F 分布中查出两个临界值 F_1 和 F_2。

若计算的 $F_i^{(l)} > F_1$，则就应把 x_i 引入方程；否则不引入。

若计算的 $F_i^{(l)} < F_2$，则应把 x_i 从回归方程中剔除，否则不剔除。

式中：$Q^{(l-1)} = 1-\sum_{i=1}^{l-1}[r_{iy}^{(i-1)}]^2 / r_{ii}^{(i-1)}$，$Q^{(l)} = 1-\sum_{i=1}^{l}[r_{iy}^{(i-1)}]^2 / r_{ii}^{(i-1)}$，$V_i = Q^{(l-1)} - Q^{(l)} = \dfrac{[r_{ly}^{(l-1)}]^2}{r_{ll}^{(l-1)}}$，$l$ 为迭代步数，n 为自变量数目，r 为相应变量的相关系数。

对于未引入回归方程的变量 x_i，逐一计算

$$V_i^{(l)} = \frac{[r_{iy}^{(l-1)}]^2}{r_{ii}^{(l-1)}}$$

再找出其中最大的一个即 V_{max}，计算

$$F_i^{(l)} = \frac{(n-l-1)V_{i\max}^{(l)}}{1-\sum_{i=1}^{l}[r_{iy}^{(i-1)}]^2 / r_{ii}^{(i-1)}}$$

如果 $F_i^{(l)} > F_1$，则引入回归方程；否则，不引入。

2. 剔除标准

对已进入回归方程的变量 x_k，逐一计算

$$V_k^{(l)}=\frac{[r_{ky}^{(l-1)}]^2}{r_{kk}^{(l-1)}}$$

找出其中最小的一个即 V_{min}，计算

$$F_k^{(l)}=\frac{(n-l-1)V_{k\max}^{(l)}}{r_{yy}^{(l)}}$$

如果 $F_k^{(l)}<F_2$，则对应变量应剔除；否则不剔除。

21.1.2　岭回归

当自变量存在高度共线性时，一般的回归分析的方差就会很大，估计值就很不稳定，有时会出现与实际意义不相符的正负号。此时可采用岭回归方法。

当自变量间存在高度共线性时，$|\boldsymbol{X'X}|\approx 0$，或者有接近于零的特征根。设想给 $\boldsymbol{X'X}$ 加上一个正常数矩阵 $\boldsymbol{KI}$（$K>0$），那么 $\boldsymbol{X'X}+\boldsymbol{KI}$ 接近奇异的程度就会比接近奇异的程度小得多。此时称

$$\widehat{\beta}(k)=(X'X+KI)^{-1}X'y$$

为 β 的岭回归，其中 k 称为岭参数，$\boldsymbol{X}$ 已经标准化，$\boldsymbol{y}$ 可以经过标准化也可以未经标准化。

显然，岭回归作为 β 的估计应比最小二乘估计稳定。当 k=0 岭回归估计就是普通的最小二乘估计。由于岭参数 k 不是唯一确定的，所以得到的岭回归估计 $\widehat{\beta}(k)$ 实际是回归参数 β 的一个估计值。当岭参数 k 在(0,∞)内变化时，$\widehat{\beta_j}(k)$ 是 k 的函数，此函数图像就称为岭迹，如图 21.2 所示。在实际应用中，可以根据岭迹曲线的变化来确定适当的值和进行自变量的选择。

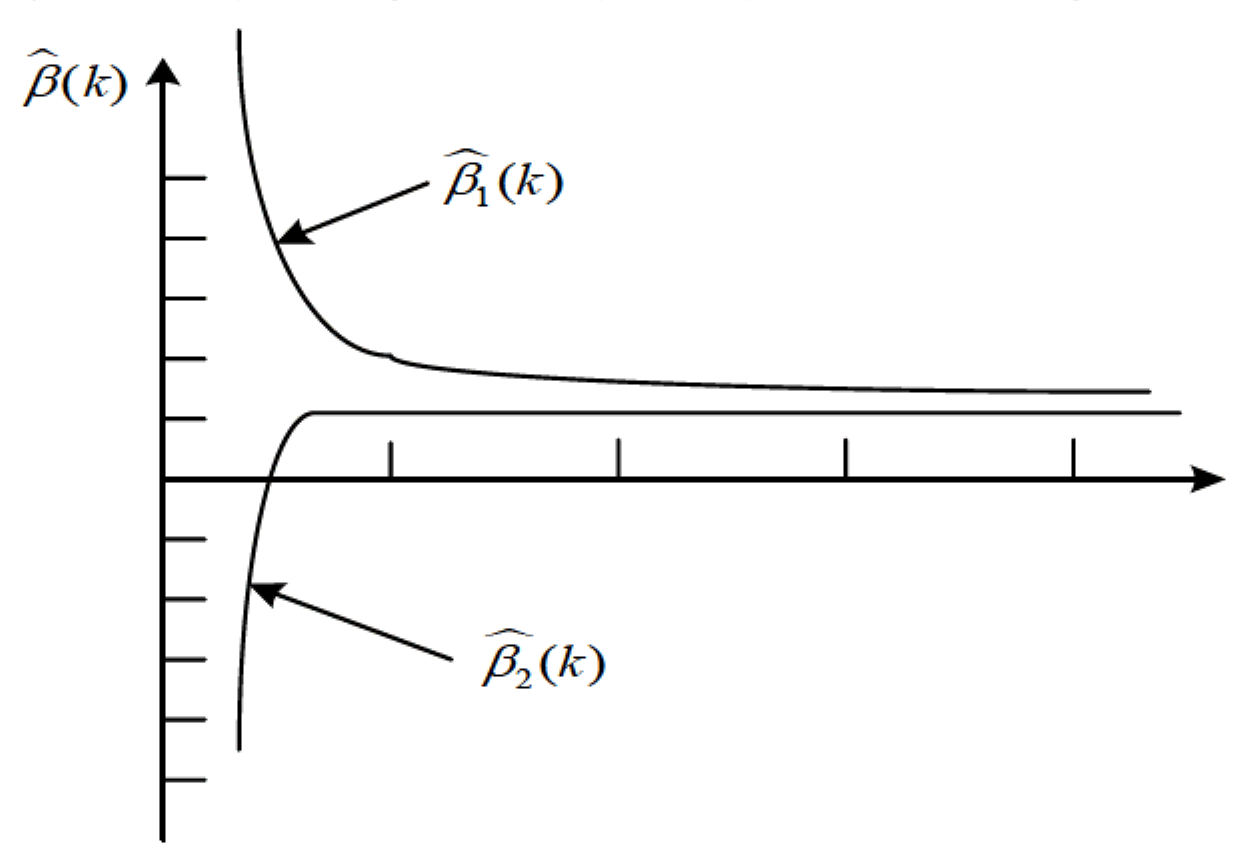


图 21.2　岭迹曲线

岭迹法选择 K 值的一般原则如下。

（1）各回归系数的岭估计基本稳定。

（2）用最小二乘估计时符号不合理的回归系统，其岭估计的符号变得合理。

（3）回归系数没有不合乎经济意义的绝对值。

（4）残差平方和增大不太多。

也可以用方差扩大因子法确定 k 值。矩阵 $c(k)=(X'X+KI)^{-1}X'X(X'X+kI)^{-1}$ 的对角线元素 $c_{jj}(k)$ 称为岭估计的方差扩大因子，其值随 k 的增大而增大，选择 k 使所有方差扩大因子 $c_{jj}(k)\leqslant 10$，此时岭估计就会变得相对稳定。

岭回归分析还可以用来选择变量，此时选择变量的原则如下。

（1）直接比较岭回归系数的大小，可以剔除回归系数比较稳定且绝对值很小的自变量。

（2）当 k 值较小时，标准化岭回归系数的绝对值并不是很小，但是不稳定，随着 k 的增加迅速趋于零，像这样岭回归系数不稳定，震动趋于零的自变量也可以剔除。

（3）如果依据上述变量的原则，有若干个回归系数不稳定，究竟去掉几个，可需要根据去掉某个变量后重新进行岭回归分析的效果来确定。

21.1.3　主成分回归分析

当自变量存在高度共线性或一般回归分析所得到的回归系数不符合常理时，可以采用主成分回归法，它通过主成分变换，将高度相关的变量的信息综合成相关性低的主成分，然后以主成分代换原变量参与回归。

主成分分析的原理及方法已在第 2 篇“数据挖掘算法”中做了介绍，而主成分回归的步骤如下。

（1）对问题的原始数据矩阵主成分分析，得到 m 个主成分 Z。

（2）然后用因变量 y，主成分 Z 作为自变量，做多元线性回归分析，得到主成分回归方程。

（3）将得到的 m 个主成分表达式代入主成分回归方程式，就会得到最终的回归方程式，即问题数据矩阵中的因变量与自变量的主成分回归方程。

21.2　时间序列预测模型

时间序列是指以时间顺序取得的一系列观察值，这里的“时间”具有广义坐标轴的含义，既可以按时间的先后顺序排列数据，也可按空间的前后顺序排列随机数据。从经济到工程技术，从天文到地理和气象，几乎在各种领域都会遇到时间序列。例如股票市场的每日波动，某地区的降水量月度序列，某化工生成过程按小时观测的产量等。

一般认为时间序列由 4 个部分构成，即：长期趋势或趋势变化，季节变动或季节性变化，循环变动或循环变化，不规则变动或随机变化。

长期趋势就是时间序列依时间变化而逐渐增加或减少的长期变化的趋势，它反映时间序列的一般变化方向。确定趋势曲线的典型方法为加权平均方法和最小二乘法。季节变化是指一年或固定一段时间内，呈现固定的规则变动，它反映每年（或固定时间段内）都重复出现的规律。循环变动主要指趋势曲线在长期时间内呈现摆动的现象，它可以是也可以不是周期性变化的。通常一个时间序列的循环是由其他多个规模小的时间序列循环组合而成的。不规则变动是在时间序列中将长期趋势、季节变动以及循环变动等成分分离后，所剩下的随机状况的部分，在数据拟合时，应先剔除不规则变动。

一般而言，长期趋势、季节变动以及循环变动都受到规则性因素的影响，可以利用一般的方法进行分析、处理和预测；而不规则变动是属于随机性的，具有不可预见性，其发生的原因很多，可能为自然灾害、人为的意外因素、天气的突然改变以及政治形势的巨大变化等。

时间序列预测（模型）就是要从历史数据中发现相似或者有规律的模式、趋势、突变以及离群点等，以揭示事物运动、变化和发展的内存规律，为人们正确认识事物和科学决策提供依据。利用时间序列模型不需要知道影响预测变量的因果关系，在系统的动态性较强、关于影响预测变量的决定性因素的信息很少、且有足够多的数据量可以用来构成一个合理长度的时间序列的情况下，运用时间序列分析往往可达到事半功倍之功效。

21.2.1 时间序列的特征量

时间序列的数学特征主要有以下几种。

1. 均值

均值的定义

$$E[x(n)] = \mu_n = \lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N}x(n)$$

对于有限长度时间序列的均值估值可按下式计算

$$E[x(n)] = \widehat{\mu_n} = \frac{1}{N}\sum_{n=0}^{N}x(n)$$

2. 方差（二阶中心矩）

方差是用来说明时间序列各可能值对其平均值的偏离程度，其定义如下

$$\sigma_x^2(x) = \sigma_x^2 = E\{|x(n)-\mu_x|^2\}$$

对于有限长度随机信号序列，计算其方差估计，可按下式

$$\sigma_x^2(x) = \sum_{n=0}^{N}\frac{1}{N}[x(n)-\mu_x]^2$$

3. 均方差

均方差定义为

$$D_x^2(x) = D_x^2 = E\{|x(n)|^2\}$$

它描述了时间序列的强度或功率。均方差与时间序列的均值和方差存在如下的关系

$$\sigma_x^2 = D_x^2 - \mu_x^2$$

21.2.2 平稳时间序列预测模型

1. 自回归模型

自回归模型记为 AR（p）。设 $x_1,x_2,\cdots,x_n$ 是平稳时间序列，则 AR（p）模型是 p 阶自回归模型，即

$$x(t)=\Phi_0+\Phi_1 x_{t-1}+\Phi_2 x_{t-2}+\cdots+\Phi_p x_{t-p}+\varepsilon_t$$

其中：t=p+1,p+2,···,n，$\Phi_p \neq 0$；ε_t 是随机误差项。

通常设 ε_t 遵从正态分布 N（0，σ_2），此时可用逐步回归法来选择 p，并得到回归系数。

2. 滑动平均模型

滑动平均模型记为 MA（q)。设

$$\begin{aligned}\tilde{x}(t)=x(n)-\overline{x}&=\varepsilon_t-\theta_1\varepsilon_{t-1}-\theta_2\varepsilon_{t-2}-\cdots\theta_q\varepsilon_{t-q}\\&=(1-\theta_1 B-\theta_2 B^2-\cdots\theta_q B^q)\varepsilon_t\end{aligned}$$

其中：$\theta_1,\theta_2,\cdots,\theta_q$ 是选定系数；$\theta_q \neq 0$，ε_t 是随机误差项，亦即白噪声。

3. 自回归—滑动模型

自回归－滑动模型记为 ARMA（p，q）。为了提高精度，满足更为一般的线性平稳模型，将 AR（p）与 MA（q）结合，组成 ARMA（p，q）模型，即自回归－滑动平均模型，其具体形式为：$\Phi(B)\tilde{x}(t)=\theta(B)\varepsilon_t$

其中：
$$\Phi(B)=1-\Phi_1 B-\Phi_2 B^2-\cdots-\Phi_p \mathrm{B}^p \qquad \Phi_p \neq 0$$

$$\theta(B)=1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q \mathrm{B}^q \qquad \theta_q \neq 0$$

以上三个模型可采用最小二乘估计法、Yule-Walker 方程估计法、U-C 算法等进行计算。

4. 平滑预测模型

（1）简单一次平滑平均预测法。

设$\{y_t\}$为时间序列，取平滑平均的项数为 n，设 y_t 是第 t 期的实际值，则第(t+1)期预测值的计算公式为

$$\hat{y}_{t+1}=M_t^{(1)}=\frac{y_t+y_{t-1}+\cdots+y_{t-n+1}}{n}=\frac{1}{n}\sum_{j=1}^{n}y_{t-n+j}$$

其中：$M_t^{(1)}$ 表示第 t 期一次平滑平均数，$\hat{y}_{t+1}$ 是第(t+1)期预测值（$t \geqslant n$），预测的标准误差为

$$S=\sqrt{\frac{\sum(y_{t+1}-\hat{y}_{t+1})^2}{N-n}}$$

其中：N 为时间序列$\{y_t\}$中原始数据的个数。

项数 n 的取值应该根据时间序列而定。如果 n 过大会降低平滑平均数的敏感性，影响预测的准确性；如果 n 过小，平滑平均数易受随机变动的影响，难以反映实际趋势。一般取的 n 值大小能包含季节变动和周期变动的时期比较好，这样可以消除它们的影响。对于没有季节变动和周期

变动的时间序列，n 的取值要视历史数据的趋势而定，一般来说，如果历史数据的类型呈水平型发展趋势，则项数 n 可取较大值；如果历史数据的类型呈上升（下降）型发展趋势，则项数 n 可取较小值，这样能取得较好的预测值。

（2）加权一次平滑平均预测法。

由于在实际中，参与平均的各期数据在预测中的作用往往是不同的，因此，需要采用加权平滑平均法进行预测。加权一次移动平均预测法是其中比较简单的一种，其计算公式为

$$\hat{y}_{t+1}=\frac{W_1y_t+W_2y_{t-1}+\cdots+W_ny_{t-n+1}}{W_1+W_2+\cdots+W_n}=\frac{\sum_{i=1}^{n}W_iy_{t-i+1}}{\sum_{i=1}^{n}W_i}$$

其中：y_t 表示第 t 期实际值；$\hat{y}_{t+1}$ 表示第(t+1)期预测值；W_i 表示权重；n 是平滑平均的项目数。

（3）一次指数平滑预测法。

一次指数平滑预测法是以 $\alpha(1-\alpha)^i$ 为权重（$0<\alpha<1$，$i=0,1,2,\cdots$），对时间序列 $\{y_t\}$ 进行加权平均的一种预测方法，y_t 的权重为 α，y_{t-1} 的权重为 $\alpha(1-\alpha)$，y_{t-1} 的权重为 $\alpha(1-\alpha)^2$，…，依次类推。计算公式为

$$\hat{y}_{t+1}=S_t^{(1)}=ay_t+(1-\alpha)S_{t-1}^1$$

其中：y_t 表示第 t 期实际值；$\hat{y}_{t+1}$ 是第 t+1 期预测值；$S_{t-1}^{(1)}$、$S_t^{(1)}$ 分别表示第 t-1 期和第 t 期的一次指数平滑值；α 表示平滑指数，$0<\alpha<1$。

预测标准误差为

$$S=\sqrt{\frac{\sum_{t=1}^{n-1}(y_{t+1}-\hat{y}_{t+1})^2}{n-1}}$$

其中：n 为时间序列中含有原始数据的个数。

平滑系数 α 对预测值有较大影响，但目前还没有一种较好的选值办法，只能根据经验来确定。当时间序列的数据呈水平型发展趋势时，α 可取较小值，通常在 0~0.3 之间；如果序列数据的类型呈上升（下降）型发展趋势，则 α 可取较大值，在 0.6~1 之间。在实际预测时，可以选取不同的 α 值进行比较，从中选取一个合适的值。

在计算指数平滑法的平滑值时，需要给出一个初值 $S_0^{(1)}$，可取原时间序列的第一项或前几项的算术平均值为初值。一次指数平滑法适用于变化比较平衡、增长或下降趋势不明显的时间序列数据预测。

（4）二次指数平滑预测法。

二次指数平滑预测法是对一次指数平滑值再作一次指数平滑来进行预测的一种方法，但第 t+1 期预测值并非第 t 期的二次指数平滑值，而是采用下列公式进行预测

$$S_t^{(2)}=aS_t^{(1)}+(1-\alpha)S_{t-1}^{(2)}$$

其中：$a_t = 2S_t^{(1)} - S_t^{(2)}$；$b_t = \dfrac{\alpha}{1-\alpha}(S_t^{(1)} - S_t^{(2)})$；$S_t^{(1)}$ 表示第期的一次指数平滑值；$S_t^{(2)}$ 表示第 t 期的二次平滑值；y_t 是第 t 期的实际值；$\hat{y}_{t+T}$ 表示第 $t+T$ 期预测值；α 是平滑系数；初值 $S_0^{(2)}$ 的取值方法与 $S_0^{(1)}$ 的取法相同。

预测的标准误差为

$$S = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n-2}}$$

二次指数平滑法适用于时间序列呈线性增长情况下的短期预测。

21.3　马尔可夫链

马尔可夫链模型是一种动态随机数学模型，它通过分析随机变量现时的运动情况来预测这些变量未来的运动情况。目前，马尔可夫链模型在自然科学、工程技术、社会科学、经济研究等领域有着广泛的应用。

设考察对象为一系统，若该系统在某一时刻可以出现的事件集合为$\{E_1,E_2,\cdots,E_N\}$，$E_1,E_2,\cdots,E_N$两两互斥，则称 E_i 为状态，$i=1,2,\cdots,N$。称该系统从一种状态 E_i 变化为另一状态 E_j 的过程为状态转移，并把整个系统不断实现状态转移的过程称为马尔可夫过程，它具有两个特点：（1）无后效性，即系统的第 n 次实际结果出现的状态，只与第 n-1 次时系统所处的状态有关，而与它以前的状态无关；（2）稳定性，该过程逐渐趋于稳定状态，与初始状态无关。

假设向量 $\boldsymbol{u}=(u_1,u_2,\cdots,u_n)$满足以下条件，则称其为概率向量

$$\begin{cases} u_j \geqslant 0, j = 1,2,\cdots,n \\ \sum_{j=1}^{n} u_j = 1 \end{cases}$$

如系统由状态 E_i 经过一次转移到状态 E_j 的概率记为 P_{ij}，则矩阵

$$P = \begin{bmatrix} P_{11} & P_{12} \cdots P_{1N} \\ P_{21} & P_{22} \cdots P_{12N} \\ \vdots & \vdots \quad\quad \vdots \\ P_{N1} & P_{N2} \cdots P_{NN} \end{bmatrix}$$

为一次（或一步）转移矩阵。

对概率矩阵 $\boldsymbol{P}$，若幂次方 $\boldsymbol{P}^m$ 的所有元素皆为正数，则矩阵 $\boldsymbol{P}$ 称为正规概率矩阵。

转移矩阵必定为概率矩阵，且具有以下的性质：

（1）$\boldsymbol{P}^{(k)}=\boldsymbol{P}^{(k-1)}\boldsymbol{P}$

（2）$\boldsymbol{P}^{(k)}=\boldsymbol{P}^k$

其中：$\boldsymbol{P}^{(k)}$为 k 次转移矩阵。

马尔可夫链模型如下：

设系统在 k=0 时的初始状态 $S^{(0)}=(S_1^{(0)},S_2^{(0)},\cdots,S_N^{(0)})$ 为已知，经过 k 次转移后的状态向量 $S^{(k)}=(S_1^{(k)},S_2^{(k)},\cdots,S_N^{(k)})$，则

$$S^{(k)}=S^{(0)}\begin{bmatrix} P_{11} & \mathrm{P}_{12}\cdots P_{1N} \\ P_{21} & \mathrm{P}_{22}\cdots P_{12N} \\ \vdots & \vdots \quad\quad \vdots \\ P_{N1} & \mathrm{P}_{N2}\cdots P_{NN} \end{bmatrix}^k$$

此式即为马尔可夫预测模型。显然，系统在经过 k 次转移后的状态 $S^{(k)}$只取决于初始状态 $S^{(0)}$和转移矩阵 $\boldsymbol{P}$。

21.4 灰色系统方法

由于人们所处的环境不同，拥有的知识水平不同，对客观世界中的许多自然现象了解程度是不一样的。按照人们对研究具体系统的了解程度，一般分为“白箱系统”“黑箱系统”和“灰箱系统”。“白箱系统”是指该系统的内部结构已被充分了解，很多情况下已经建立了该系统的数学模型；“黑箱系统”则是指那些系统内部结构一点都不被了解，只能获取该系统的激励与响应信息，有的甚至这些信息都很难获取；则“灰箱系统”是介于“白箱系统”与“黑箱系统”之间，即知道系统的一些简单信息，但是并没有完全了解该系统，只能根据统计推断或某种逻辑思维来研究该系统，研究的方法即为灰色系统方法。

21.4.1 灰色系统的基本概念

由于自然现象的复杂性，人们不可能对所有的自然系统都有充分的了解，必定存在许多灰色系统甚至黑色系统。很明显对于灰色系统的描述有别于白色系统。

1. 灰数

灰色系统理论中的一个重要概念是灰数。灰数是灰色系统理论的基本单元。人们把只知道大概范围而不知道其确切值的数称为灰数。在应用中，灰数实际上指在某一个区间或某个一般的数集内取值的不确定数。灰数是区间数的一种推广，通常用记号“⊗”表示。

有以下几类灰数：

- 仅有下界的灰数。有下界而无上界的灰数，记为$\otimes\in[\underline{a},\infty)$，其中 $\underline{a}$ 为灰数的下确界，它是一个确定的数，$[\underline{a},\infty)$称为⊗的取数域，简称⊗的灰域。
- 仅有上界的灰数。有上界而无下界的灰数记为$\otimes\in(\infty,\ \overline{a}]$，其中 $\overline{a}$ 为灰数的上确界，是一个确定的数，而$(\infty,\ \overline{a}]$是它的灰域。
- 区间灰数。既有上界又有下界的灰数称为区间灰数，记为$\otimes\in[\underline{a},\ \overline{a}]$。
- 连续灰数与离散灰度。在某一个区间内取有限个值或可数个值的灰数称为离散灰数；取值连续地充满某一区间的灰数称为连续灰数。
- 黑数与白数。当$\otimes\in(-\infty,\infty)$或$\otimes\in((\otimes_1,\otimes_2)$，即当⊗的上、下界皆为无穷或上、下界都是灰数时，称⊗为黑数，可见，黑数是上、下界都不确定的数。当$\otimes\in[\underline{a},\ \overline{a}]$

且 $\underline{a}=\overline{a}$ 时，称⊗为白数，即取值为确定的值。可以把白数和黑数看成是特殊的灰数。

- 本征灰数与非本征灰数。本征灰数是指不能或暂时还不能找到一个白数作为其“代表”的灰数，比如一般的事前预测值。非本征灰数是指凭先验信息或某种手段，可以找到一个白数作为其代表的灰数。此白数称为相应灰数的白化值。记为 $\tilde{\otimes}$，并用⊗（a）表示以 a 为白化值的灰数。

从本质上看，灰数又可以分为信息型、概念型和层次型三类。信息型灰数是指由于信息缺乏而不能肯定其取值的数；概念型灰数是由人们的某种意愿、观念形成的灰数；层次型灰数是由层次改变而形成的灰数。

2. 灰数白化与灰度

当灰数是在某个基本值附近变动的，这类灰数白化比较容易，可以其基本值 a 为主要白化值，记为⊗（a）$=a\pm\delta_a$ 或⊗（a）∈（$-,a,+$），其中 δ_a 为扰动灰元，此灰数的白化值为 $\tilde{\otimes}(a)=a$。

对于一般的区间灰数 $\otimes\in[a,b]$，将白化值 $\tilde{\otimes}$ 取为

$$\tilde{\otimes}=\alpha a+(1-\alpha)b,\alpha\in[0,1]$$

也可称为等权白化。在等权白化中，取 α=1/2 而得到的白化值称为等权均值白化值。当区间灰数取值的分布信息缺乏时，常采用等权均值白化。

一般而言，灰数的白化取决于信息的多少，如信息量较大则白化较为容易。一般用白化权函数（α 即为权）来描述一个灰数对其取值范围内不同数值的“偏爱”程度。一个灰数的白化权函数是研究者根据已知信息设计的，没有固定的格式。

灰度即为灰数的测度。灰数的灰度在一定程度上反映了人们以灰色系统的行为特征的未知程度。一个灰数的灰度大小应与该灰数产生的背景或论域有关不可分割的作用。在实际应用中，会遇到大量的白化权函数未知的灰数。灰数的灰度主要与相应定义信息域的长度及其基本值有关。

21.4.2　灰色序列生成算子

灰色系统理论的主要任务之一是根据社会、经济、生态等系统的行为特征数据，寻找不同系统变量之间或某些系统变量自身的数学关系和变化规律。灰色系统理论认为任何随机过程都是在一定幅度范围内和一定时区内变化的灰色量，并把随机过程看成灰色过程。

由于受到噪声的干扰，需要采用统计的方法研究给定的某一数据序列。但是统计的方法要求数据量非常大，并且计算量大，也无法对动态数据的发展趋势进行预测，尤其是对小样本数据，统计方法更显得力不从心。灰色系统可以克服上述缺憾，它利用一定的数据处理方法去寻找数据间的发展演变规律。

灰色系统理论通过对原始数据的挖掘（预处理），生成新的数据序列，以便挖掘出原始数据中的规律，发现隐匿在数据中的趋势，这样一种以数据寻找数据现实规律的途径被称为灰色序列生成。灰色系统认为，尽管客观系统表象复杂，数据离乱，但它总是有整体功能的，因而必然蕴含某种内在规律，关键在于如何选择适当的方式去挖掘它和利用它。一切灰色序列都能通过某种生成弱化其随机性，显现其规律性。

设 $X=(x(1),x(2),\cdots,x(n))$为原始数据序列，$D$ 为作用于 X 的算子，X 经过算子 D 的作用后所得的序列为：$XD=(x(1)d,x(2)d,\cdots,x(n)d)$。称 D 为序列算子，称 XD 为一阶算子作用序列。

且 $\underline{a}=\overline{a}$ 时，称⊗为白数，即取值为确定的值。可以把白数和黑数看成是特殊的灰数。

- 本征灰数与非本征灰数。本征灰数是指不能或暂时还不能找到一个白数作为其“代表”的灰数，比如一般的事前预测值。非本征灰数是指凭先验信息或某种手段，可以找到一个白数作为其代表的灰数。此白数称为相应灰数的白化值。记为 $\tilde{\otimes}$，并用⊗（a）表示以 a 为白化值的灰数。

从本质上看，灰数又可以分为信息型、概念型和层次型三类。信息型灰数是指由于信息缺乏而不能肯定其取值的数；概念型灰数是由人们的某种意愿、观念形成的灰数；层次型灰数是由层次改变而形成的灰数。

2. 灰数白化与灰度

当灰数是在某个基本值附近变动的，这类灰数白化比较容易，可以其基本值 a 为主要白化值，记为⊗（a）$=a\pm\delta_a$ 或⊗（a）∈（$-,a,+$），其中 δ_a 为扰动灰元，此灰数的白化值为 $\tilde{\otimes}(a)=a$。

对于一般的区间灰数 $\otimes\in[a,b]$，将白化值 $\tilde{\otimes}$ 取为

$$\tilde{\otimes}=\alpha a+(1-\alpha)b,\alpha\in[0,1]$$

也可称为等权白化。在等权白化中，取 $\alpha=1/2$ 而得到的白化值称为等权均值白化值。当区间灰数取值的分布信息缺乏时，常采用等权均值白化。

一般而言，灰数的白化取决于信息的多少，如信息量较大则白化较为容易。一般用白化权函数（α 即为权）来描述一个灰数对其取值范围内不同数值的“偏爱”程度。一个灰数的白化权函数是研究者根据已知信息设计的，没有固定的格式。

灰度即为灰数的测度。灰数的灰度在一定程度上反映了人们以灰色系统的行为特征的未知程度。一个灰数的灰度大小应与该灰数产生的背景或论域有关不可分割的作用。在实际应用中，会遇到大量的白化权函数未知的灰数。灰数的灰度主要与相应定义信息域的长度及其基本值有关。

21.4.2 灰色序列生成算子

灰色系统理论的主要任务之一是根据社会、经济、生态等系统的行为特征数据，寻找不同系统变量之间或某些系统变量自身的数学关系和变化规律。灰色系统理论认为任何随机过程都是在一定幅度范围内和一定时区内变化的灰色量，并把随机过程看成灰色过程。

由于受到噪声的干扰，需要采用统计的方法研究给定的某一数据序列。但是统计的方法要求数据量非常大，并且计算量大，也无法对动态数据的发展趋势进行预测，尤其是对小样本数据，统计方法更显得力不从心。灰色系统可以克服上述缺憾，它利用一定的数据处理方法去寻找数据间的发展演变规律。

灰色系统理论通过对原始数据的挖掘（预处理），生成新的数据序列，以便挖掘出原始数据中的规律，发现隐匿在数据中的趋势，这样一种以数据寻找数据现实规律的途径被称为灰色序列生成。灰色系统认为，尽管客观系统表象复杂，数据离乱，但它总是有整体功能的，因而必然蕴含某种内在规律，关键在于如何选择适当的方式去挖掘它和利用它。一切灰色序列都能通过某种生成弱化其随机性，显现其规律性。

设 $X=(x(1),x(2),\cdots,x(n))$ 为原始数据序列，D 为作用于 X 的算子，X 经过算子 D 的作用后所得的序列为：$XD=(x(1)d,x(2)d,\cdots,x(n)d)$。称 D 为序列算子，称 XD 为一阶算子作用序列。

序列算子可以作用多次，相应得到的序列称为二阶、三阶序列……相应的算子称为一阶、二阶序列算子……

1. 均值生成算子

在收集数据时，常常由于一些不易克服的困难导致数据序列出现空缺（即空穴）；而有些数据序列虽然完整，但由于系统行为在某个时点上发生突变而形成异常数据，剔除异常数据后就会留下空穴。

如何填补序列空穴自然成为数据处理过程中首先遇到的问题，均值生成是常用的构造新数据，填补原序列空穴，生成新序列的方法。

设序列在 k 处出现空穴，记为 $\varnothing(k)$，即

$$X=(x(1),x(2),\cdots,x(t\text{-}1),\varnothing(k),x(t+1)\cdots,x(n))$$

称 $x(t\text{-}1)$ 和 $x(t+1)$ 为 $\varnothing(t)$ 的界值，前者为前界，后者为后界。

当 $\varnothing(k)$ 是则 $x(t-1)$ 和 $x(t+1)$ 生成时，称生成值 $x(t)$ 为 $[x(t-1),x(t+1)]$ 的内点。

而当 $\varnothing(k)=x^*(t)=0.5\,x(t-1)+0.5\,x(t)$ 称为非紧邻均值生成数。

设序列 $X=(x(1),x(2),\cdots,x(n),x(n+1))$，$Z$ 是 X 的均值生成序列

$$Z=(z(1),z(2),\cdots,z(n))$$

其中：$z(t)=0.5\,x(t-1)+0.5\,x(t)$，$X^*$ 是某一可导函数的代表序列，d 为 n 维空间的距离，将 X 删除 $x(n+1)$ 后提到的序列仍记为 X，若 X 满足

① 当充分大时，$x(t)<\sum_{i=1}^{t\text{-}1}x(i)$

② $\max_{l\leqslant t\leqslant n}|x^*(t)-x(t)|\geqslant\max_{l\leqslant t\leqslant n}|x^*(t)-z(t)|$

则称 X 为光滑序列。称 $\rho(t)=\dfrac{x(t)}{\sum_{i=1}^{t\text{-}1}x(i)}$　$t=2,3,\cdots,n$ 为 X 的光滑比。

2. 累加生成算子

累加生成可以看出灰量积累过程的发展趋势，使杂乱的原始数据中蕴含的积分特性或规律充分表现出来。

设 $X^0=(x^0(1),x^0(2),\cdots,x^0(n))$，D 为序列算子，即：$X^0D=(x^0(1)d,x^0(2)d,\cdots,x^0(n)d)$

其中：$x^0(t)=\sum_{i=1}^{t}x^0(i)$　　$t=1,2,\cdots,n$

则称 D 为 X^0 的一次累加算子，记为 1-AGO。同样可以有二阶、三阶、……、r 阶的累加生成算子，可以记为 $x^r(t)d=\sum_{i=1}^{t}x^{r-1}(i)$　　$t=1,2,\cdots,n$

则累加生成算子生成的序列称为累加生成数。

如果原始序列为非负准光滑序列，则其一次累加生成序列具有准指数性质。原始序列越光滑，生成后指数规律也越明显。

3. 累减生成算子

设 $X^0=(x^0(1),x^0(2),\cdots,x^0(n))$，$D$ 为序列算子，即：$X^0D=(x^0(1)d,x^0(2)d,\cdots,x^0(n)d)$
其中：$x^0(k)=x^0(t)-x^0(t-1)$　　$t=1,2,\cdots,n$，规定 $x^{(1)}(0)=0$
则称 D 为 X^0 的一次累减算子，记为 1-IAGO。同样可以有二阶、三阶、……、r 阶的累减生成算子。

由累减生成算子生成的序列称为累减生成数。

21.4.3　灰色分析

灰色系统建模是通过数据序列建立微分方程来拟合给定的时间序列，从而对数据的发展趋势进行预测。

灰色建模常用的模型是 GM（1,N），其中 G 代表灰色，1 代表微分方程的阶数，N 代表变量的个数。

1. GM（1,1）模型

给定数列　$X^0=(x^0(1),x^0(2),\cdots,x^0(n))$

$X^1=(x^1(1),x^1(2),\cdots,x^1(n))$

$Z^1=(Z^1(1),Z^1(2),\cdots,Z^1(n))$

方程 $x^0(k)+a\ Z^1(k)=b$ 为灰微分方程，$-a$ 为发展灰数，反映了序列的发展趋势；b 为内生控制灰数，它反映了数据变化的关系，其确切内涵是灰色的。

其中：$x^0(k)$为原始数据序列；X^1 为 X^0 的 1-AGO 序列；$Z^1(k)=0.5\,x^1(k)+0.5\,x^1(k-1)$为 X^1 的近邻生成序列。

设 $\hat{a}=(a,b)$ 为参数列，令

$$Y=\begin{bmatrix} x^0(2) \\ x^0(3) \\ \vdots \\ x^0(n) \end{bmatrix}, B=\begin{bmatrix} -z^1(2) & 1 \\ -z^1(3) & 1 \\ \vdots & \vdots \\ -z^1(n) & 1 \end{bmatrix}$$

则灰微分方程 $x^0(k)+a\,Z^1(k)=b$ 的最小二乘估计参数列满足

$$\hat{a}=(B^TB)^{-1}B^TY$$

给定数列　$X^0=(x^0(1),x^0(2),\cdots,x^0(n))$，$X^1$ 为 X^0 的 1-AGO 序列，Z^1 为 X^1 的紧邻生成序列，称

$$\frac{dx^{(1)}}{dt}+ax^{(1)}=b$$

为灰微分方程的白化方程，也称影子方程；其解

$$x^{(1)}(t)=(x^{(1)}(0)-\frac{b}{a})e^{-at}+\frac{b}{a}$$

称为时间响应函数。

GM（1,1）灰微分方程 $x^0(t)+a\,Z^1(t)=b$ 的时间响应序列为

$$\hat{x}^{(1)}(t+1)=(x^{(1)}(0)-\frac{b}{a})e^{-at}+\frac{b}{a} \qquad t=1,2,\cdots,n$$

取 $x^{(1)}(0)=x^{(1)}(1)$，则

$$\hat{x}^{(1)}(t+1)=(x^{(1)}(1)-\frac{b}{a})e^{-at}+\frac{b}{a} \qquad t=1,2,\cdots,n$$

还原值为

$$\hat{x}^{(0)}(t+1)=x^{(1)}(t+1)-\hat{x}^{(1)}(t) \qquad t=1,2,\cdots,n$$

通过大量的实际问题验证，对于 GM（1,1）的使用范围如下。

- 当 $-a\leqslant 0.3$ 时，可用于中长期预测；
- 当 $0.3<-a\leqslant 0.5$ 时，可用于短期预测，中长期预测慎用；
- 当 $0.5<-a\leqslant 0.8$ 时，作短期预测应十分谨慎；
- $0.8<-a\leqslant 1$ 时，应采用残差修正 GM（1,1）模型；
- 当 $-a>1$ 时，不宜采用 GM（1,1）模型。

2. GM（1,1）模型检验

GM（1,1）模型的检验有残差检验、关联度检验和后验差检验。

（1）残差检验。

残差大小检验是对模型值与实际值的残差进行逐点检验。

绝对残差序列

$$\Delta^{(0)}=\{\Delta^{(0)}(i),i=1,2,\cdots,n\},\ \Delta^{(0)}(i)=|\Delta^{(0)}(i)-\hat{x}^{(0)}(i)|$$

及相对残差序列

$$\phi=\{\phi_i,i=1,2,\cdots,n\},\ \phi_i=|\frac{\Delta^{(0)}(i)}{x^{(0)}(i)}]\%$$

并计算相对残差

$$\bar{\phi}=\frac{1}{n}\sum_{i=1}^{n}\phi_i$$

给定 α，当 $\bar{\phi}<\alpha$,且 $\phi_n<\alpha$ 成立时，称模型为残差检验合格模型。

（2）关联度检验。

关联度检验是通过考察模型值曲线和建模序列曲线的相似程度进行检验。按前面所述的关联度计算方法，计算出 $\hat{x}^{(0)}(i)$ 与原始数列 $x^{(0)}(i)$ 的关联系数，然后计算出关联度。根据经验，关联度大于 0.6 是可以接受的。

（3）后验差检验。

后验差检验是对残差分布的统计特性进行检验。

①计算出原始数列的平均值

$$\overline{x}^{(0)} = \frac{1}{n}\sum_{i=1}^{n} x^{(0)}(i)$$

②计算原始数列的均方差

$$S_1 = \left(\frac{\sum_{i=1}^{n} [x^{(0)}(i) - \overline{x}^{(0)}]^2}{n-1} \right)^{\frac{1}{2}}$$

③计算残差的均值

$$\overline{\Delta} = \frac{1}{n}\sum_{i=1}^{n} \Delta^{(0)}(i)$$

④计算残差的方差

$$S_2 = \left(\frac{\sum_{i=0}^{n} [\Delta^{(0)}(i) - \overline{\Delta}]^2}{n-1} \right)^{\frac{1}{2}}$$

⑤计算方差比

$$C=S_1/S_2$$

⑥计算小残差概率

$$P = P\{|\Delta^{(0)}(i) - \overline{\Delta}| < 0.6745S_1$$

令 $S_0 = 0.6745\text{S}_1$，$e_i = |\Delta^{(0)}(i) - \overline{\Delta}|$，即 $P = P\{e_i < S_0\}$。

若对于给定的 $C_0 > 0$，当 $C < C_0$时，称模型为均方差比合格模型。如对于给定的 $P_0 > 0$，当 $P > P_0$称为小残差概率合格模型。

若相对残差、关联度、后验差检验在允许的范围内，则可以用所建立的模型进行预测，否则应进行残差修正。

3. 残差 GM（1,1）模型

当 GM（1，N）模型的精度不符合要求时，可以用参差序列建立 GM（1,N）模型对原来的模型进行修正，以提高精度。

设 $X^0=(x^0(1),x^0(2),\cdots,x^0(n))$为模型的原始序列，$X^1$ 为 X^0 的 1-AGO 序列，Z^1 为 X^1 的紧邻生成序列，灰色微分方程 $x^0(t)+a Z^1(t)=b$ 的时间响应序列为

$$\hat{x}^{(1)}(t+1) = (x^{(1)}(0) - \frac{b}{a})e^{-at} + \frac{b}{a} \qquad t = 1,2,\cdots,n$$

其参差序列为

$$\varepsilon^{(0)} = (\varepsilon^{(0)}(1), \varepsilon^{(0)}(2), \cdots, \varepsilon^{(0)}(n))$$

其中：$\varepsilon^{(0)}(t) = x^{(1)}(t) - \hat{x}^{(1)}(t)$，若存在，满足

- 对任意的 $t \geqslant t_0, \varepsilon^{(0)}(t)$，符号一致；
- $n- t_0 \geqslant 4$，则称

$$(|\varepsilon^{(0)}(t_0)|,|\varepsilon^{(0)}(t_0+1)|,\cdots,|\varepsilon^{(0)}(n)|)$$

为可建模参差尾段，仍记为

$$\varepsilon^{(0)}=(\varepsilon^{(0)}(t_0),\varepsilon^{(0)}(t_0+1),\cdots,\varepsilon^{(0)}(n))$$

对于可建模参差尾段，其 $1-\text{AGO}$ 序列

$$\varepsilon^{(1)}=(\varepsilon^{(1)}(k_0),\varepsilon^{(1)}(k_0+1),\cdots,\varepsilon^{(1)}(n))$$

的 GM（1，1）时间响应式为

$$\hat{\varepsilon}^{(1)}(t+1)=\left(\varepsilon^{(0)}(t_0)-\frac{b_\varepsilon}{a_\varepsilon}\right)e^{-a(t-t_0)}+\frac{b_\varepsilon}{a_\varepsilon}$$

则参差尾段的模拟序列为：$\hat{\varepsilon}^{(0)}=(\hat{\varepsilon}^{(0)}(t_0),\hat{\varepsilon}^{(0)}(t_0+1),\cdots,\hat{\varepsilon}^{(0)}(n))$，其中

$$\hat{\varepsilon}^{(0)}(t+1)=-a_\varepsilon\left(\varepsilon^{(0)}(t_0)-\frac{b_\varepsilon}{a_\varepsilon}\right)e^{-a(t-t_0)},t \geqslant t_0$$

若用 $\varepsilon^{(0)}(k)$ 修正 $\widehat{X}^{(1)}$，称修正后的时间响应式

$$\hat{x}^{(1)}(t+1)=\begin{cases}\left(x^{(0)}(1)-\dfrac{b}{a}\right)e^{-at}+\dfrac{b}{a} & t<t_0\\ \left(x^{(0)}(1)-\dfrac{b}{a}\right)e^{-at}+\dfrac{b}{a}\pm a_\varepsilon\left(\varepsilon^{(0)}(t_0)-\dfrac{b_\varepsilon}{a_\varepsilon}\right)e^{-a_\varepsilon(t-t_0)} & t \geqslant t_0\end{cases}$$

为参差修正 GM（1,1）模型。

4. GM（1,N）模型

设 $X_1^0=(x_1^0(1),x_1^0(2),\cdots,x_1^0(n))$为系统特征数据序列，$X_i^0=(x_i^0(1),x_i^0(2),\cdots,x_i^0(n))$，$i=2,3,\cdots,N$ 为相关因素数列序列，对 X_i^0 作累加生成 $X_i^{(1)}$，称为 X_i^0 的阶累生成序列

$$x_i^{(0)}(t)=\sum_{m=1}^{t}x_i^{(0)}(m),\qquad t=1,2,\cdots,n;i=1,2,\cdots,N$$

$$X_i^{(1)}=(x_i^{(1)}(1),x_i^{(1)}(2),\cdots,x_i^{(1)}(n))\qquad i=1,2,\cdots,N$$

$Z_1^{(1)}$ 为 $X_1^{(1)}$ 的紧邻均值生成序列，建立如下形式的微分方程模型

$$\frac{dx_1^{(1)}(t)}{dt}+az_1^{(1)}(t)=b_1x_2^{(1)}(t)+b_2x_3^{(1)}(t)+\cdots+b_{n-1}x_N^{(1)}(t)$$

为是一阶 N 个变量的微分方程模型，称为 GM(1,N)模型。

利用最小二乘法对该方程求解，可得到系数阵

$$\hat{a}=(B^T B)^{-1}B^T Y$$

其中：$\hat{a}=(a,b_1,\cdots,b_{n-1})^T$；$y=[x_1^{(0)}(2),x_1^{(0)}(3),\cdots,x_1^{(0)}(N)]^T$

$$B=\begin{bmatrix} -\frac{1}{2}[z_1^{(1)}(1)+z_1^{(1)}(2)] & \frac{1}{2}[x_2^{(1)}(1)+x_2^{(1)}(2)] & \cdots & \frac{1}{2}[x_N^{(1)}(1)+x_N^{(1)}(2)] \\ -\frac{1}{2}[z_1^{(1)}(2)+z_1^{(1)}(3)] & \frac{1}{2}[x_2^{(1)}(2)+x_2^{(1)}(3)] & \cdots & \frac{1}{2}[x_N^{(1)}(2)+x_N^{(1)}(3)] \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{1}{2}[z_1^{(1)}(n-1)+z_1^{(1)}(n)] & \frac{1}{2}[x_2^{(1)}(n-1)+x_2^{(1)}(n)] & \cdots & \frac{1}{2}[x_N^{(1)}(n-1)+x_N^{(1)}(n)] \end{bmatrix}$$

模型建立后，通过求解微分方程得 $\widehat{x_1}^{(1)}(t)$，并将其作累减得模型还原值 $\widehat{x_1}^{(0)}(t)$，并与实测原始值比较，看是否满足精度要求，若否，对残差继续建立 GM 模型进行修正。

5. 灰色灾变预测

灰色灾变预测的任务是给出下一个或几个异常值出现的时刻，以便人们提前准备，采取对策，减少损失。

设原始数列为 $\boldsymbol{X}=\{x(1),x(2),\cdots,x(n)\}$，给定上限异常值（灾变值）ζ，称 X 的子序列

$$\boldsymbol{X}=\{x(q(1)),x(q(2)),\cdots,x(q(m))\}=\{x(q(i))|x(q(i))\geqslant\zeta \quad i=1,2,\cdots,m\}$$

为上灾变序列。

如果给定下限异常值（灾变值）ξ，则称 X 的子序列

$$\boldsymbol{X}=\{x(q(1)),x(q(2)),\cdots,x(q(l))\}=\{x(q(i))|x(q(i))\leqslant\xi \quad i=1,2,\cdots,l\}$$

为下灾变序列。

如原始序列 $\boldsymbol{X}_\zeta=\{x(q(1)),x(q(2)),\cdots,x(q(m))\}\subset X$ 为灾变序列，相应的数列 $Q^{(0)}=\{q(1),q(2),\cdots,q(m)\}$ 为灾变日期序列。

对于灾变日期序列，其 1－AGO 序列为 $Q^{(1)}=\{q(1),q(2),\cdots,q(m)\}$ 的紧邻生成序列 $Z^{(1)}$，则 $q(t)+a Z^1(t)=b$ 为灾变 GM（1,1）模型。

设 $\boldsymbol{\alpha}=[a,b]^{\mathrm{T}}$ 为灾变 GM（1,1）模型参数序列的最小二乘估计，则灾变日期序列的 GM（1,1）序号响应式为

$$\hat{q}^{(1)}(t+1)=(q(1)-\frac{b}{a})e^{-at}+\frac{b}{a}$$

$$\hat{q}(t+1)=\hat{q}^{(1)}(t+1)-\hat{q}^{(1)}(t)$$

$$\hat{q}(t+1)=(q(1)-\frac{b}{a})e^{-at}-(q(1)-\frac{b}{a})e^{-a(t-1)}=(1-e^{a})(q(1)-\frac{b}{a})e^{-at}$$

设 $\boldsymbol{X}=\{x(1),x(2),\cdots,x(n)\}$ 为原始数列，n 为现在，给定异常值 ζ，相应的灾变日期序列 $Q^{(0)}=\{q(1),q(2),\cdots,q(m)\}$，其中 $q(m)\leq n$ 为最近一次灾变日期，则称 $\hat{q}(m+1)$ 为下一次灾变的预测日期，对任意 $t>0$，称 $\hat{q}(m+t)$ 为未来第 t 次灾变的预测日期。

21.5 例题

例 4.18　表 21.1 是我国在某段时间内财政收入的数据表。请对其进行回归分析。

表 21.1　财政收入表

国民收入（亿元）	工业总产值（亿元）	农业总产值（亿元）	总人口（百万）	就业人口（百万	固定资产投资（亿元）	财政收入（亿元）
598	349	461	57482	20729	44	184
586	455	475	58796	21364	89	216
707	520	491	60266	21832	97	248
737	558	529	61465	22328	98	254
825	715	556	62828	23018	150	268
837	798	575	64653	23711	139	286
1028	1235	598	65994	26600	256	357
1114	1681	509	67207	26173	338	444
1079	1870	444	66207	25880	380	506
757	1156	434	65859	25590	138	271
677	964	461	67295	25110	66	230
779	1046	514	69172	26640	85	266
943	1250	584	70499	27736	129	323
1152	1581	632	72538	28670	175	393
1322	1911	687	74542	29805	212	466
1249	1647	697	76368	30814	156	352
1187	1565	680	78534	31915	127	303
1372	2101	688	80671	33225	207	447
1638	2747	767	82992	34432	312	564
1780	3156	790	85229	35620	355	638
1833	3365	789	87177	35854	354	658

解：

如果对此问题进行一般的多元线性回归分析，得到的某些回归系数为负值，明显不符合财政收入与各个指标间的实际关系。这说明各指标之间具有较强的相关性。此时的回归分析需要采用岭回归。

MATLAB 中有专门的岭回归函数 ridge。为了应用方便，对此函数进行了改进，主要通过判断回归系数的稳定性，即连续 n 次回归系数的差不超过某个值（er），就为较好的 k 值。用户也可以通过图 21.3 所示的“岭迹图”确定较佳 k 值，然后输入带 k 值的 mybridge 函数进行回归。利用得到的回归系数进行回归时，不需要对数据进行规范化处理。

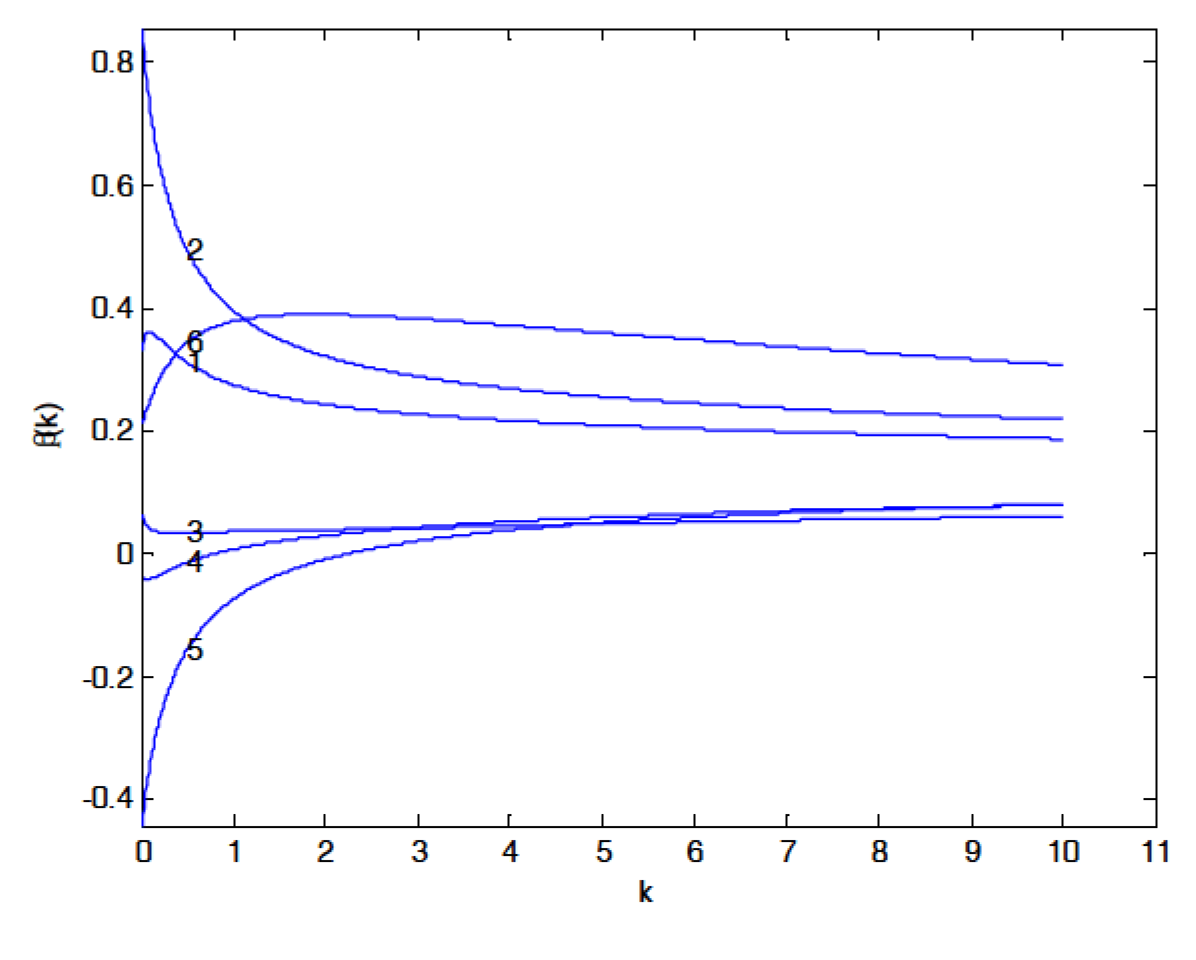


图 21.3　岭迹图

n、er 以及最大的 k 值可以采用默认值，也可以自行输入。

```
>>load mydata;
>> [a,b]=mybridge(x,y)
a=80.03640.09850.0602 0.0443  0.0002  -0.0015   0.5049        %回归系数,包括常数项
b=1.23                                                         %较好的 k 值
```

从岭迹图可以看出变量 x_3、x_4 和 x_5 的回归系数较小，可以忽略这三个变量的影响。然后对原始样品删除这三个变量后再进行回归分析。

例 4.19　在回归分析的实际应用中，应注意由于测量仪器性能、外界条件等因素的影响，得到的数据集有可能存在异常或粗差值，或者各自变量对因变量测量误差的影响程度并不相同。在这些情况下，回归分析应采用稳健回归，即采用含权重参数的回归模型

$$(X^{\mathrm{T}}X+wI)\ \mathbf{b}=X^{\mathrm{T}}\mathbf{y}$$

式中：$\boldsymbol{X}$ 为测量数据矩阵；$\boldsymbol{y}$ 为响应值矩阵；$\boldsymbol{b}$ 为估计得到的回归系数；$\boldsymbol{w}$ 为权重，它是一个可调的正数；I 为单位矩阵。

表 21.2 是婴儿年龄（月）、身高与体重关系的数据集，请对此进行回归分析。

表 21.2　数据集

年　龄	身　高	体　重	年　龄	身　高	体　重	年　龄	身　高	体　重
112	141.6	37.6	128	134.0	30.3	134	154.5	52.3
116	147.8	42.8	129	148.5	45.5	135	152.0	50.5
117	142.8	40.5	129	146.3	41.6	137	151.5	49.4
120	140.7	39.5	130	147.5	42.2	139	150.6	48.5
123	134.7	34.3	131	158.8	59.3	140	149.9	47.5
125	145.4	38.0	132	132.0	49.0	141	160.3	59.3
126	135.0	32.5	133	148.7	43.5			

解：

```
>>x1=[112 116 117 120 123 125 126 128 129 129 130 131 132 133 134 135 137 139 140 141];
  x2=[141.6 147.8 142.8 140.7 134.7 145.4 135.0 134.0 148.5 146.3 147.5 158.8 132.0
148.7 154.5 152.0 151.5 150.6 149.9 160.3];
  x3=[37.6 42.8 40.5 39.5 34.3 38.0 32.5 30.3 45.5 41.6 42.2 59.3 49.0 43.5 52.3
50.5 49.4 48.5 47.5 59.3];
>>num=length(x1); x=[x1;x2]';
>> b=regress(x3',[ones(num,1),x]);            %一般线性回归分析
>> bb=b(2:3)';y=b(1)+bb*x';plot3(x1,x2,y1,'*'); hold on;plot3(x1,x2,x3,'o');  %图21.4
>> b=robustfit(x,x3');                          %稳健回归分析
>>bb=b(2:3)';y2=b(1)+bb*x';plot3(x1,x2,x3,'o');hold on;plot3(x1,x2,y2,'*')   %图21.5
```

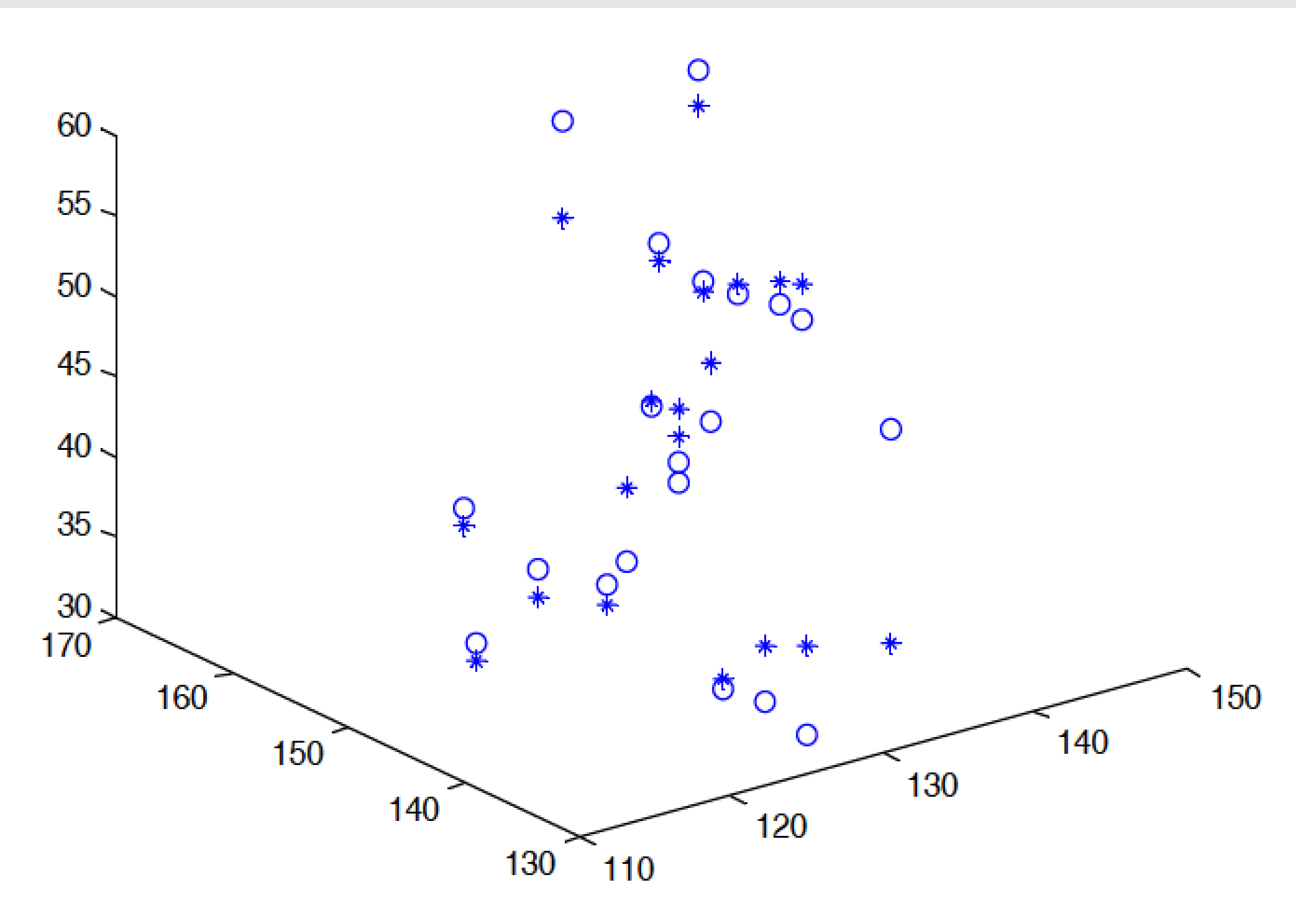


图 21.4　一般线性回归结果

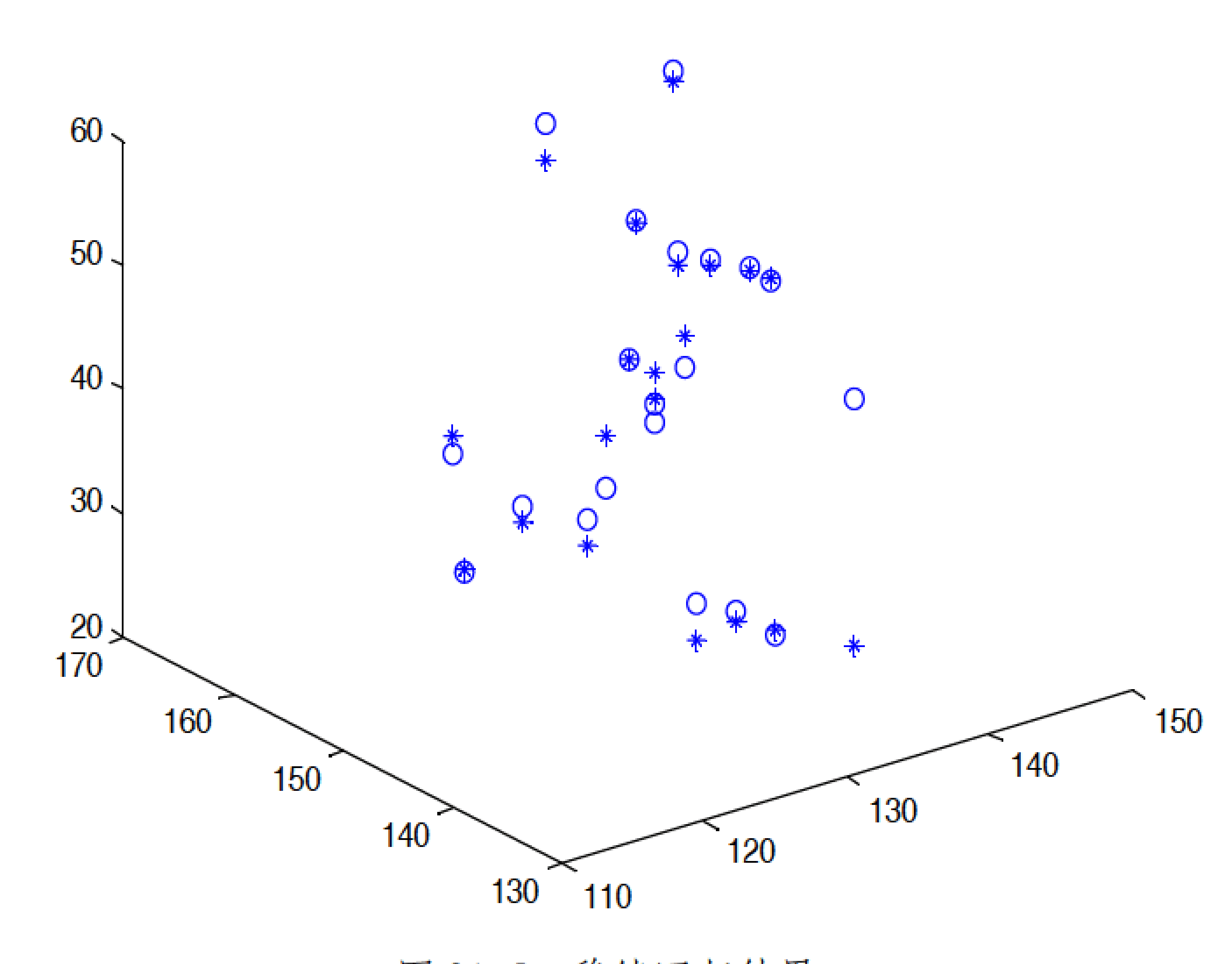


图 21.5　稳健回归结果

从图中可看出，稳健回归的结果要明显好于一般线性回归结果，这主要是由于数据集中有异常点（12 号样本）存在。

例 4.20　在实际问题中，经常遇到需要研究两组多重相关变量间的相互依赖关系，并研究用一组变量（常称为自变量或预测变量）去预测另一组变量（常称为因变量或响应变量），除了最小二乘准则下的经典多元线性回归分析（MLR），提取自变量组主成分的主成分回归分析（PCR）等方法外，还有近年发展起来的偏最小二乘（PLS）回归方法。偏最小二乘回归提供一种多对多线性回归建模的方法，特别当两组变量的个数很多，且都存在多重相关性，而观测数据的数量（样本量）又较少时，用偏最小二乘回归建立的模型具有传统的经典回归分析等方法所没有的优点。

偏最小二乘回归分析在建模过程中集中了主成分分析，典型相关分析和线性回归分析方法的特点，因此在分析结果中，除了可以提供一个更为合理的回归模型外，还可以同时完成一些类似于主成分分析和典型相关分析的研究内容，提供更丰富、深入的一些信息。

表 21.3 是某健身俱乐部的 20 位中年男子的一些体能指标。一组是身体特征指标 X，包括体重、腰围、脉搏。第二组是训练结果指标 Y，包括单杠、弯曲、跳高。

表 21.3　体能训练数据

体 重（x_1）	腰 围（x_2）	脉 搏（x_3）	单 杠（y_1）	弯 曲（y_2）	跳 高（y_3）
191	36	50	5	162	60
189	37	52	2	110	60
193	38	58	12	101	101
162	35	62	12	105	37
189	35	46	13	155	58
182	36	56	4	101	42
211	38	56	8	101	38
167	34	60	6	125	40
176	31	74	15	200	40
154	33	56	17	251	250
169	34	50	17	120	38
166	33	52	13	210	115
154	34	61	14	215	105
247	46	50	1	50	50
193	36	46	6	70	31
202	37	62	12	210	120
176	37	54	4	60	25
157	32	52	11	230	80
156	33	54	15	225	73
138	33	68	2	110	43

解：

对于偏最小二乘回归，既可以自己编程，也可以用MATLAB自带的偏最小二乘函数plsregress进行计算。在此利用自编的pls进行求解。

从图中可看出，稳健回归的结果要明显好于一般线性回归结果，这主要是由于数据集中有异常点（12 号样本）存在。

例 4.20　在实际问题中，经常遇到需要研究两组多重相关变量间的相互依赖关系，并研究用一组变量（常称为自变量或预测变量）去预测另一组变量（常称为因变量或响应变量），除了最小二乘准则下的经典多元线性回归分析（MLR），提取自变量组主成分的主成分回归分析（PCR）等方法外，还有近年发展起来的偏最小二乘（PLS）回归方法。偏最小二乘回归提供一种多对多线性回归建模的方法，特别当两组变量的个数很多，且都存在多重相关性，而观测数据的数量（样本量）又较少时，用偏最小二乘回归建立的模型具有传统的经典回归分析等方法所没有的优点。

偏最小二乘回归分析在建模过程中集中了主成分分析，典型相关分析和线性回归分析方法的特点，因此在分析结果中，除了可以提供一个更为合理的回归模型外，还可以同时完成一些类似于主成分分析和典型相关分析的研究内容，提供更丰富、深入的一些信息。

表 21.3 是某健身俱乐部的 20 位中年男子的一些体能指标。一组是身体特征指标 X，包括体重、腰围、脉搏。第二组是训练结果指标 Y，包括单杠、弯曲、跳高。

表 21.3　体能训练数据

体 重 (x_1)	腰 围 (x_2)	脉 搏 (x_3)	单 杠 (y_1)	弯 曲 (y_2)	跳 高 (y_3)
191	36	50	5	162	60
189	37	52	2	110	60
193	38	58	12	101	101
162	35	62	12	105	37
189	35	46	13	155	58
182	36	56	4	101	42
211	38	56	8	101	38
167	34	60	6	125	40
176	31	74	15	200	40
154	33	56	17	251	250
169	34	50	17	120	38
166	33	52	13	210	115
154	34	61	14	215	105
247	46	50	1	50	50
193	36	46	6	70	31
202	37	62	12	210	120
176	37	54	4	60	25
157	32	52	11	230	80
156	33	54	15	225	73
138	33	68	2	110	43

解：

对于偏最小二乘回归，既可以自己编程，也可以用MATLAB自带的偏最小二乘函数plsregress进行计算。在此利用自编的pls进行求解。

```
>>load mydata;
>> [sol,r,rr]=pls(x,y);                     %得图21.6、图21.7
>> sol=47.0197   612.5671  183.9849         %回归系数
       -0.0167   -0.3509    -0.1253
       -0.8237   -10.2477   -2.4969
       -0.0969   -0.7412    -0.0518
 r=2                                        %主成分数
```

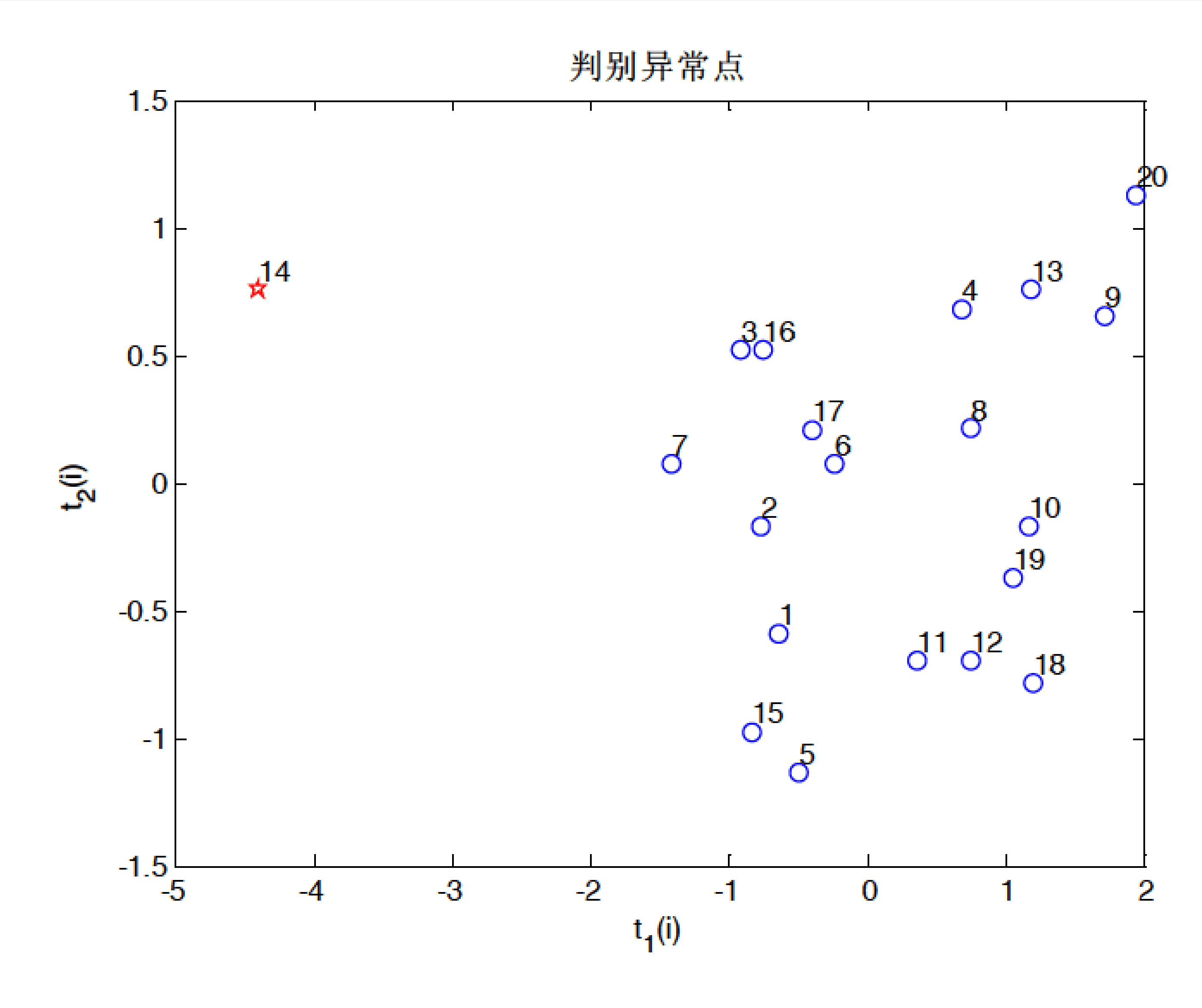


图 21.6 异常点判断

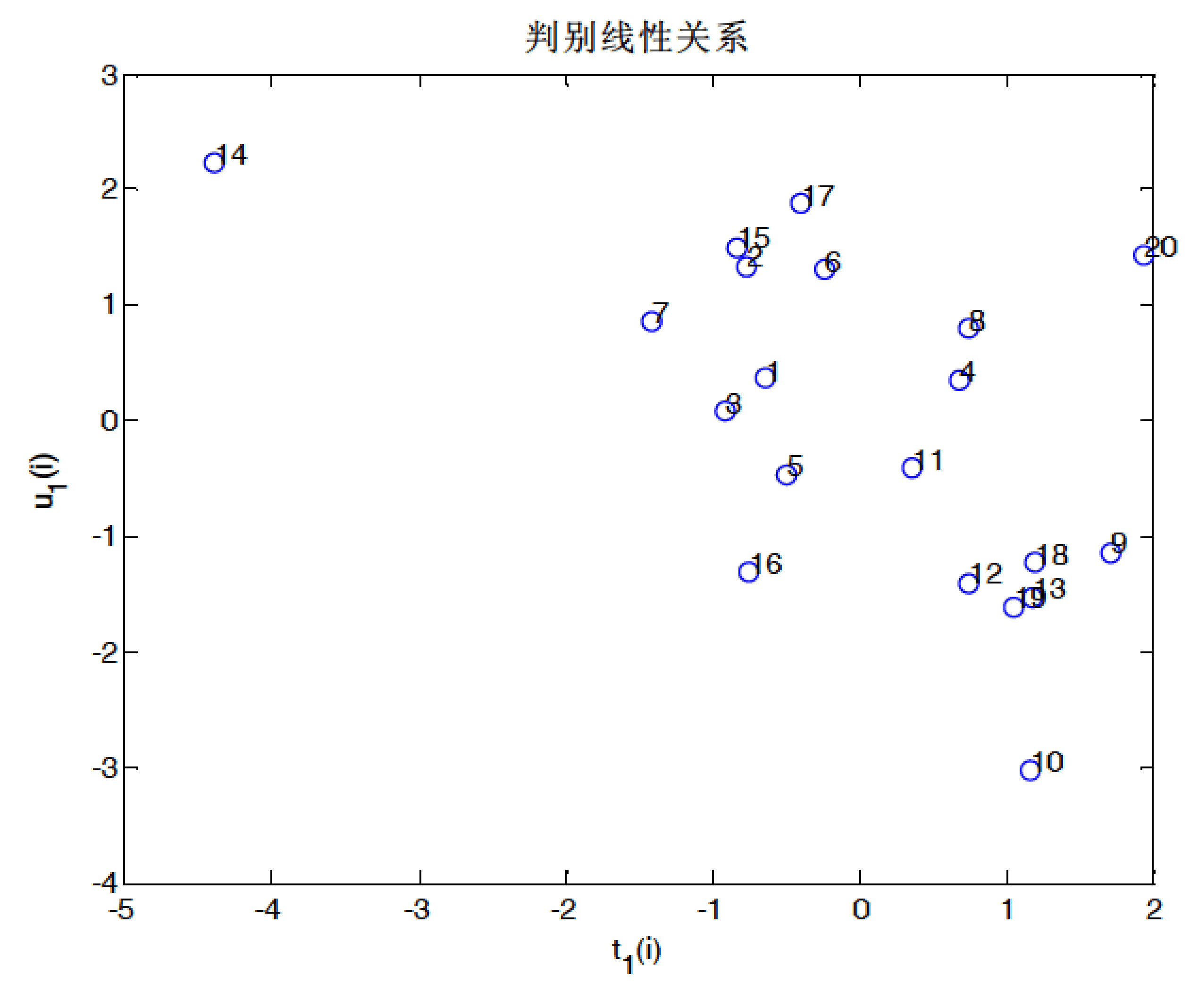


图 21.7 线性关系判断

从图中可看出，这个问题的线性关系不明显，预测结果的误差较大。另外14号样本点属于异常点。

例 4.21　在回归分析的实际中，经常会遇到多指标的问题。多指标不仅造成计算复杂，而且它们之间可能存在的相关性使它们提供的整体信息发生重叠，不易得出简单的规律。解决变量间的这个多重共线性问题，除了应用偏最小二乘外，还可以使用主成分分析。主成分分析中将多指标问题转化成较少的综合指标问题，综合指标是原来多个指标的线性组合，虽然这些线性综合指标不能观测到，但这些综合指标间互不相关，又能反映原来多指标的信息。

表 21.4 是一个数据集，试用主成分回归方法对其回归分析。

表 21.4　数据集

x_1	x_2	x_3	x_4	x_5	y
15.57	2463	472.92	18	4.45	566.52
44.02	2048	1339.75	9.5	6.92	696.82
20.42	3940	620.25	12.8	4.28	1033.15
18.74	6505	568.33	36.7	3.9	1603.62
49.2	5723	1497.6	35.7	5.5	1611.37
44.92	11520	1365.83	24.0	4.6	1613.27
55.48	5779	1687.0	43.3	5.62	1854.17
59.28	5969	1639.92	46.7	5.15	2160.55
94.39	8461	2872.33	78.7	6.18	2305.58
128.02	20106	3055.08	180.5	6.15	3503.93
96.0	13313	2912.0	60.9	5.88	3571.89
131.42	10771	3921.0	103.7	4.88	3741.4
127.21	15543	3865.67	126.8	5.5	4026.52
252.9	36194	7684.1	157.7	7.0	10343.81
409.2	34703	12446.33	169.4	10.78	11732.17
463.7	39204	14098.4	331.4	7.05	15414.94
510.2	86533	15524	371.6	6.35	18854.45

解：

```
>>load mydata;
>> [sol,pc1,pcNum]=prinregress(x,y);    %得图 21.8
sol=-727.9139 8.0614  0.0698  0.2629   13.7414  104.2156
pcNum=2;
```

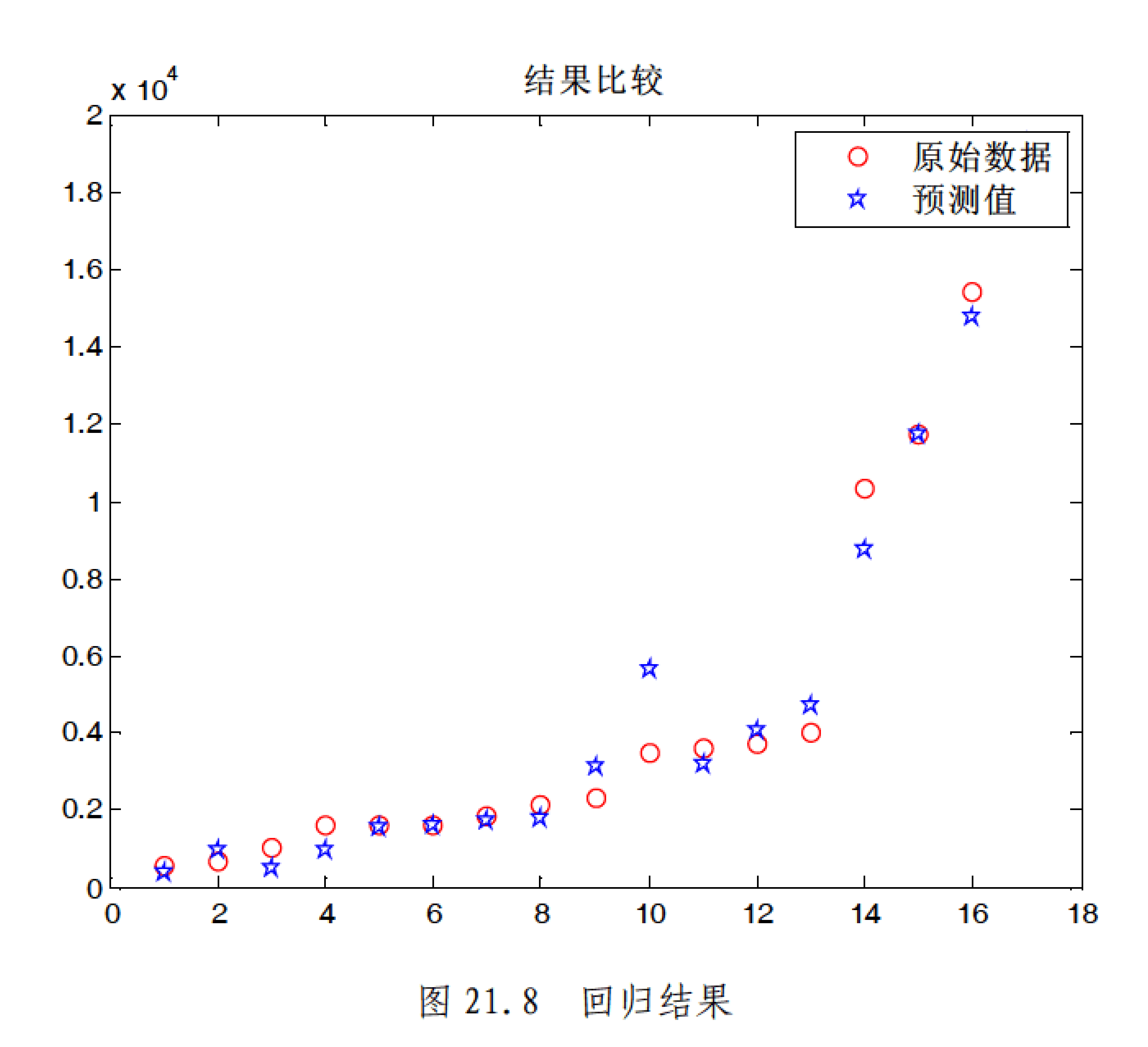


图 21.8 回归结果

例 4.22 某地区的经济发展情况如表 21.5 所示。请用主成分回归、岭回归、偏最小二乘方法对其分析。

表 21.5 经济情况数据

x_1	149.3	161.2	171.5	175.5	180.8	190.7	202.1	212.4	226.1	231.9	239.0
x_2	4.2	4.1	3.1	3.1	1.1	2.2	2.1	5.6	5.0	5.1	0.7
x_3	108.1	114.8	123.2	126.9	132.1	137.7	146.0	154.1	162.3	164.3	167.6
Y	15.9	16.4	19.0	19.1	18.8	20.4	22.7	26.5	28.1	27.6	26.3

解：

```
 >> x=[149.3 161.2 171.5 175.5 180.8 190.7 202.1 212.4 226.1 231.9 239.0;
       4.2    4.1   3.1    3.1   1.1    2.2    2.1   5.6    5.0    5.1    0.7;
       108.1 114.8 123.2 126.9 132.1 137.7 146.0 154.1 162.3 164.3 167.6]';
     y=[15.9 16.4  19.0  19.1  18.8  20.4  22.7  26.5  28.1  27.6  26.3]';
>> [sol,r,rr,yy]=pls(x,y);                      %图 21.9
>> [sol,pc1,pcNum]=prinregress(x,y);            %图 21.10
>> [k_b,beta]=mybridge(x,y);                    %图 21.11
```

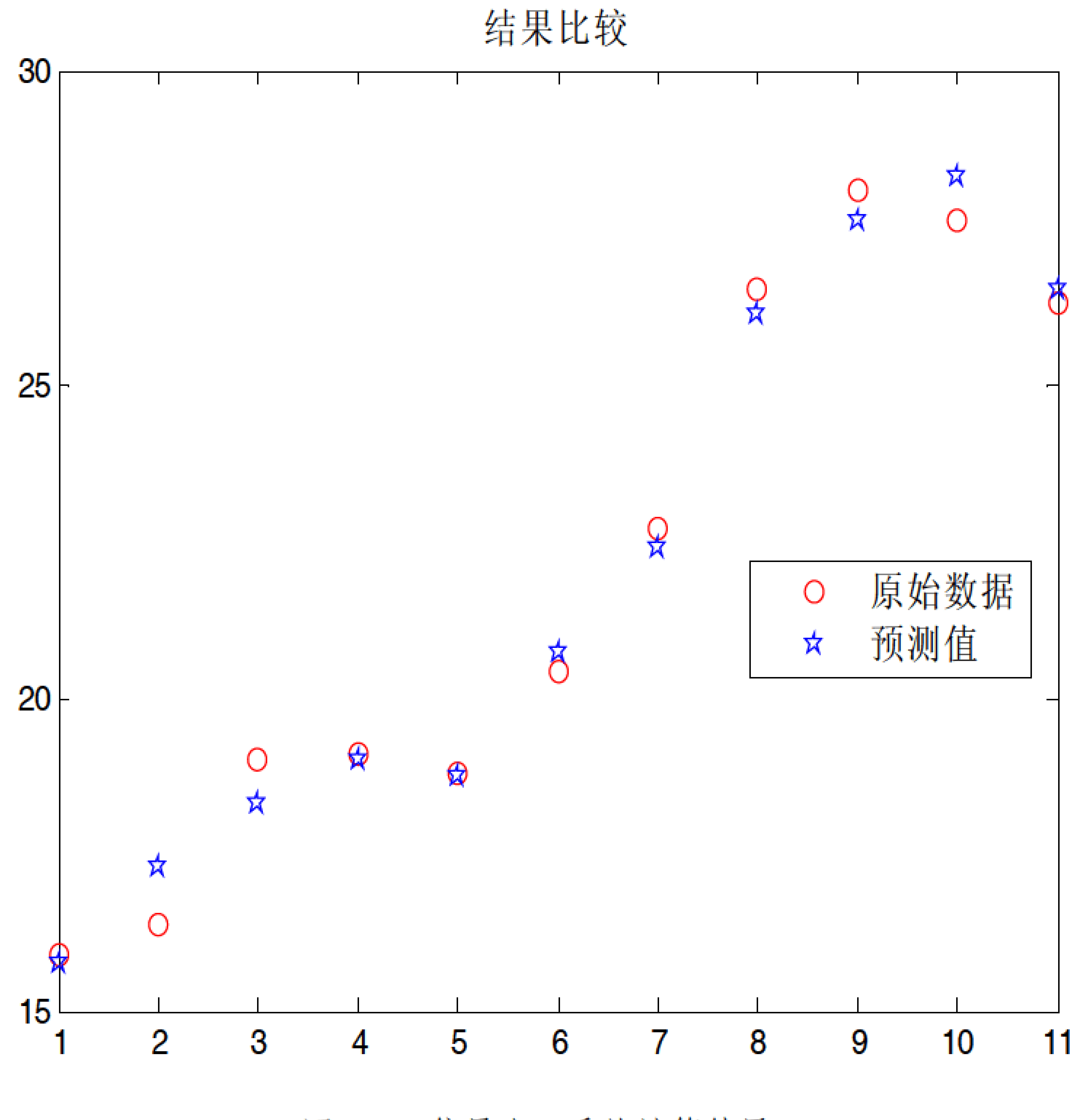



图 21.9 偏最小二乘的计算结果

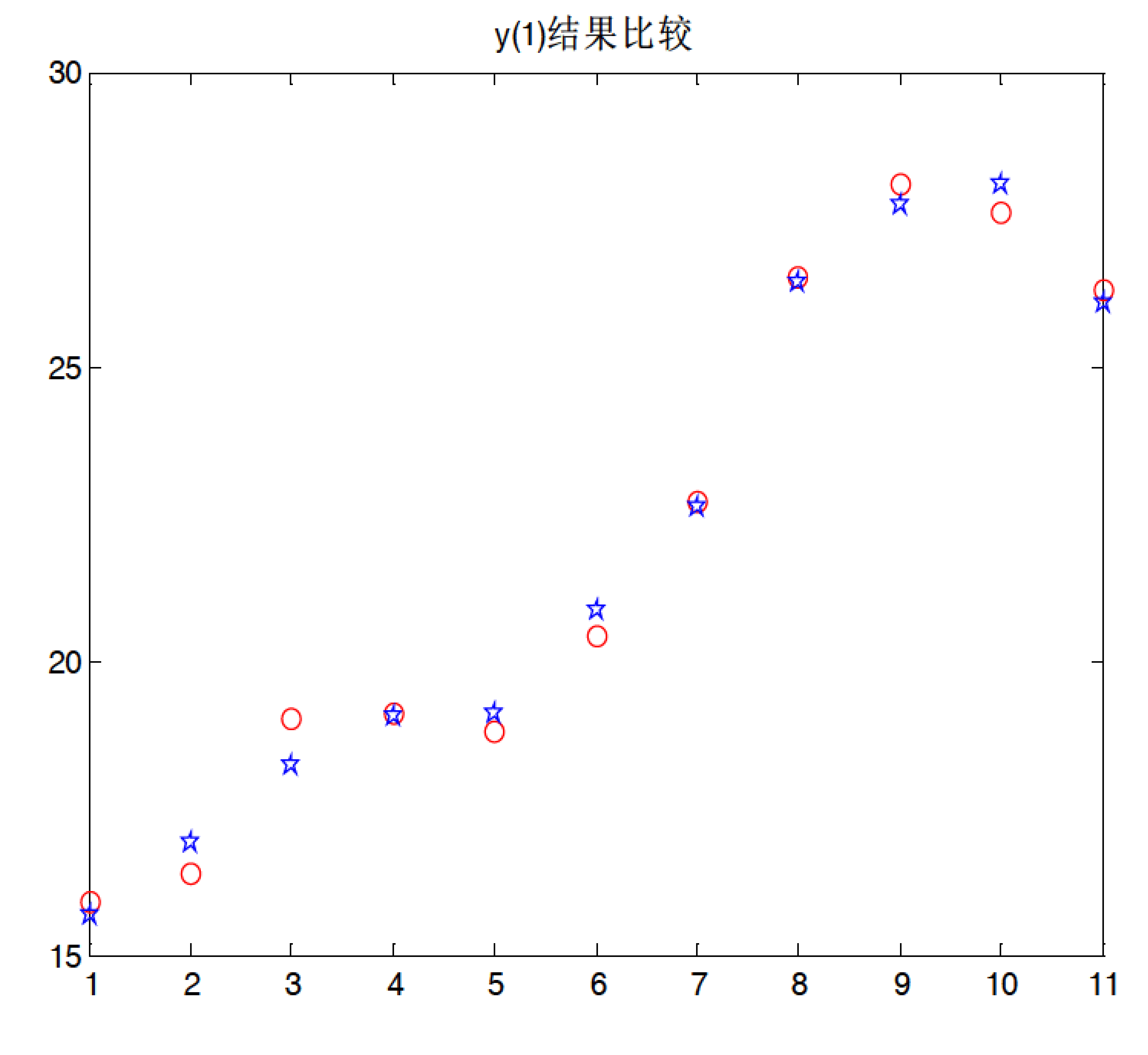


图 21.10　主成分回归的计算结果

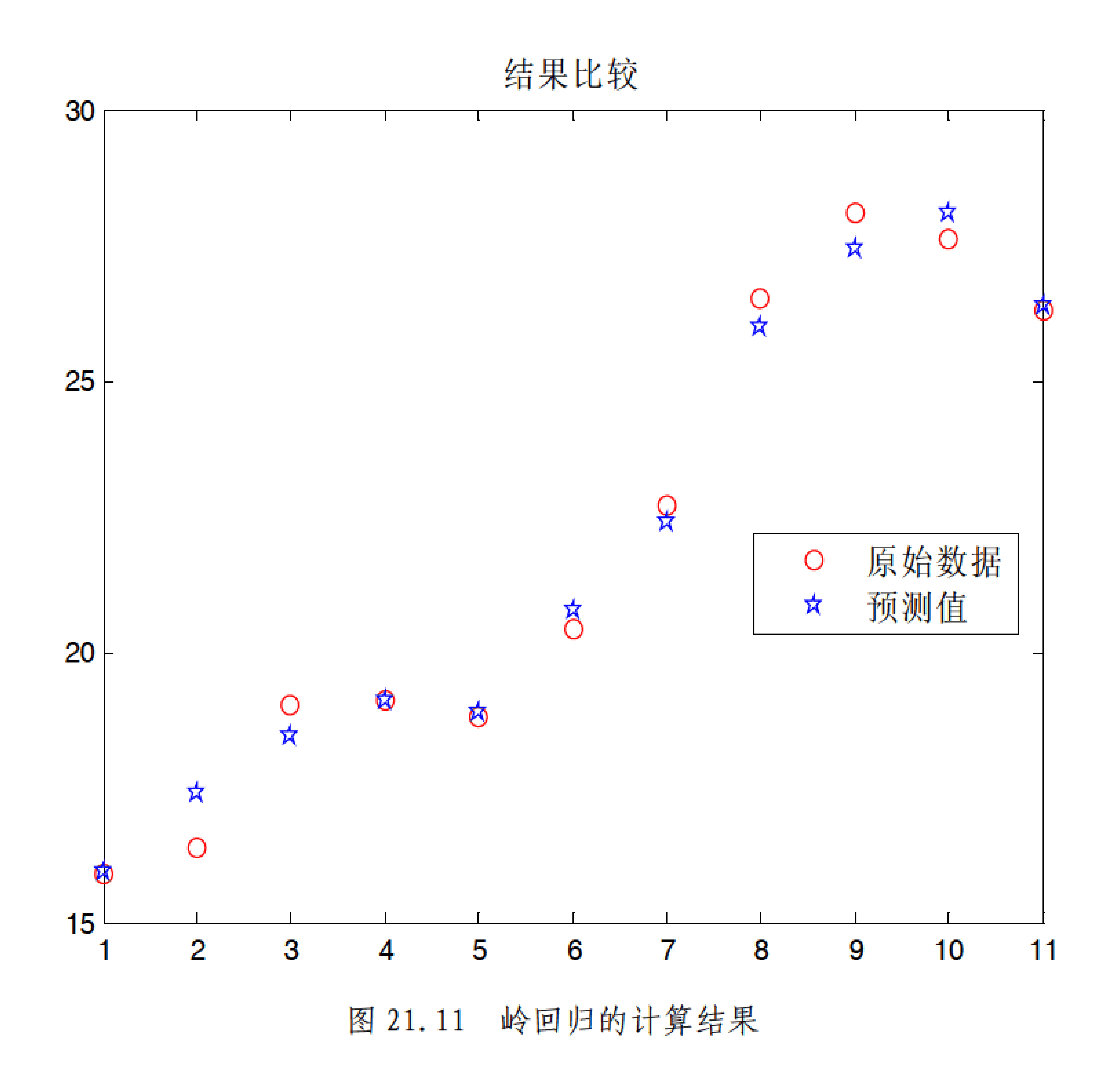


图 21.11　岭回归的计算结果

从计算结果的图示中可看出，三者之间偏最小二乘的计算结果最好。

例 4.23　在回归分析中，除了应用偏最小二乘、逐步回归、主成分回归等回归方法外，还可以对变量进行增删，以得到合适的回归表达式。

某钢铁公司炼钢转炉的炉龄按 30 天炉/天炼钢规模，大约一个月就需等炉一次进行检修。为了减少消耗，厂家通过实际测定，得到表 21.6 所示的数据，其中 x_1 为喷补料量、x_2 为吹炉时间、x_3 为炼钢时间、x_4 为钢水中含锰量、x_5 为渣中含铁量、x_6 为作业率、目标变量 y 为炉龄（炼钢炉次/炉）。试根据此表数据建立炉龄的预测模型，以便适当调节参数，以延长炉龄。

表 21.6　转炉炉龄数据

No.	x_1	x_2	x_3	x_4	x_5	x_6	y
1	0.2922	18.5	41.4	58.0	18.0	83.3	1030
2	0.2672	18.4	41.0	51.0	18.0	91.7	1006
3	0.2685	17.7	38.6	52.0	17.3	78.9	1000
4	0.1835	18.9	41.8	18.0	12.8	47.2	702
5	0.2348	18.0	39.4	51.0	17.4	57.4	1087
6	0.1386	18.9	40.5	39.0	12.8	22.5	900
7	0.2083	18.3	39.8	64.0	17.1	52.6	708
8	0.4180	18.8	41.0	64.0	16.4	26.7	1223
9	0.1030	18.4	39.2	20.0	12.3	35.0	803
10	0.4893	19.3	41.4	49.0	19.1	31.3	715
11	0.2058	19.0	40.0	40.0	18.8	41.2	784

续表

No.	x_1	x_2	x_3	x_4	x_5	x_6	y
12	0.0925	17.9	38.7	50.0	14.3	66.7	535
13	0.1854	19.0	40.8	44.0	21.0	28.6	949
14	0.1963	18.1	37.2	46.0	15.3	63.0	1012
15	0.1008	18.2	37.0	46.0	16.8	33.9	716
16	0.2702	18.9	39.5	48.0	20.2	31.3	858
17	0.1465	19.1	38.6	45.0	17.8	28.1	826
18	0.1353	19.0	38.6	42.0	16.7	39.7	1015
19	0.2244	18.8	37.7	40.0	17.4	49.0	861
20	0.2155	20.2	40.2	52.0	16.8	41.7	1098
21	0.0316	20.9	41.2	48.0	17.4	52.6	580
22	0.0491	20.3	40.6	56.0	19.7	35.0	573
23	0.1487	19.4	39.5	42.0	18.3	33.3	832
24	0.2445	18.2	36.6	41.0	15.2	37.9	1076
25	0.2222	18.4	37.0	40.0	13.7	42.9	1376
26	0.1298	18.4	37.2	45.0	17.2	44.3	914
27	0.2300	18.4	37.1	47.0	22.9	21.6	861
28	0.2436	17.7	37.2	45.0	16.2	37.9	1105
29	0.2804	18.3	37.5	46.0	17.3	20.3	1013
30	0.1970	17.3	35.9	46.0	13.8	57.4	1249
31	0.1840	16.2	35.3	43.0	16.6	44.8	1039
32	0.1679	17.1	34.6	43.0	20.3	37.3	1502
33	0.1524	17.6	36.0	51.0	14.2	36.7	1128

解：

变量的增减可以用遗传算法来完成。变量扩维—筛选方法可分为两个步骤。

① 变量扩维：将含有变量 $x_1,x_2,\cdots,x_n$ 的数据矩阵 $\boldsymbol{X}$ 扩维，引入变量的非线性项如 $x_1^2, x_2^2\cdots$，$x_1x_2,\cdots,x_1/x_2$ 和其他函数形式的项，这样将 $\boldsymbol{X}$ 扩维到 $\boldsymbol{X}'$。在这个过程中，宁可多增加一些变量，也不要遗漏变量。

② 从矩阵 $\boldsymbol{X}'$的变量筛选出一些重要的变量，或最佳变量组合形成的矩阵 X'来建立模型，使得所建立的模型有较强或最好的预报能力。

变量扩维较为简单，关键是变量筛选。变量筛选问题，特别是当变量的数目比较大时，是十分复杂的问题。解决这个问题可以采用多种方法，其中遗传算法是其中的一种。

在处理变量筛选问题时，遗传算法的编码一般采用二进制编码。对变量数为 n 的问题，可用一个含有 n 个 0 或 1 的字符串表示一个变量组合，1 和 0 分别表示此变量选中和未选中，1 在字符串的位置表示变量的序号。如“00110110”，表示有 8 个变量，其中第 3、4、6 和 7 变量被选中。

编码结束后，再利用一般的遗传算法的基本步骤，就可以求出最佳个体，即变量数及含义，此时所采用的适应度函数为适应度函数用 PRESS 值。此值的含义如下：将 m 样本中 m-1 个样本

用作训练样本，剩下的一个样本做检验样本。利用 m-1 样本建模，用检验样本代入模型，可求得一个估计值 y_1。然后换另外一个样本作为检验样本，用其余样本建模，检验样本检验，得到第二个估计值 y_2。如此循环 m 次，每次都留下一个样本作估计，最后可求得 m 个估计值，并可求出 m 个预报残差 y_i-y_{i-1}，再将这 m 个残差平方求和，即为 PRESS。此值越小，表示模型的预报能力越强。

$$\text{PRESS}=\sum_{i=1}^{m}(y_i-y_{i-1})^2$$

为了减少计算量，在实际中可以通过普通残差来求 PRESS，即

$$\text{PRESS}=\sum_{i=1}^{m}(\frac{e_i}{1-h_{ii}})^2$$

式中：e_i 为普通残差；h_{ii} 为第 i 个样本点到样本点中心的广义化距离；$h_{ii}=\boldsymbol{x}^{\mathrm{T}}_i(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{x}_i$。$\boldsymbol{X}$ 为数据矩阵，$\boldsymbol{x}_i$ 为 $\boldsymbol{X}$ 中的某一行矢量。

具体对本例来说，除表中的变量外，还可以加上如表 21.7 所示变量：

表 21.7 变量

变 量	非线性因子	变 量	非线性因子	变 量	非线性因子
x_7	x_1^2	x_{14}	x_2x_3	x_{21}	x_3x_6
x_8	x_1x_2	x_{15}	x_2x_4	x_{22}	x^2_4
x_9	x_1x_3	x_{16}	x_2x_5	x_{23}	x_4x_5
x_{10}	x_1x_4	x_{17}	x_2x_6	x_{24}	x_4x_6
x_{11}	x_1x_5	x_{18}	x^2_3	x_{25}	x^2_5
x_{12}	x_1x_6	x_{19}	x_3x_4	x_{26}	x_5x_6
x_{13}	x_2^2	x_{20}	x_3x_5	x_{27}	x^2_6

据此，可编程计算，得到以下的结果。

```
>>load data;
>> y1=selectvar(data,y);
   y1= 0     0     1     2     2     3     5     1     2     4
       3     4     4     3     4     5     6     1     2     4
```

即 x_3，x_4，x_1x_4，x_2x_3，x_2x_4，x_3x_5，x_5x_6，x_1^2，x_2^2，x^2_4 这些变量被选中。

例 4.24 表 21.8 为我国在 1965—1984 年期间的发电总量，试预测下一年的发电总量。

表 21.8 发电量资料

年 份	发 电 量	年 份	发 电 量
1965	676	1976	2031
1966	825	1977	2234
1967	774	1978	2566
1968	716	1979	2820
1969	940	1980	3006
1970	1159	1981	3093

续表

年　份	发 电 量	年　份	发 电 量
1971	1384	1982	3277
1972	1524	1983	3514
1973	1668	1984	3770
1974	1688		
1975	1958		

解：

用于时间序列预测的平滑方法可以用多种选择，在此选择二次指数平滑法。

```
>> x=[676 825 774 716 940 1159 1384 1524 1668 1688 1958 2031 2234 2566 2820 3006
    3093 3277 3514 3770];
>> [y,a,b]=smoothpre(x,'qE',1);
>> y=3.9166e+003              %预测值（实际值4107）
```

smoothpre函数所采用的预测方法有移动平均法（简单移动平均法、加权移动平均法、趋势移动平滑法）、指数平滑法（一次、二次及三次指数）和差分指数平滑法（一阶差分和二阶差分）。

简单移动平均法和加权移动平均法，在时间序列没有明显的趋势变动时，能够准确反映实际情况。但当时间序列出现直线增加或减少的变动趋势时，用简单移动平均法和加权移动平均法来预测就会出现滞后偏差。因此，需要进行修正，修正的方法是作二次移动平均，利用移动平均滞后偏差的规律来建立直线趋势的预测模型，这就是趋势移动平均法。

一般说来历史数据对未来值的影响是随时间间隔的增长而递减的。所以，更切合实际的方法应是对各期观测值依时间顺序进行加权平均作为预测值。指数平滑法可满足这一要求，而且具有简单的递推形式。

但当时间序列的变动具有直线趋势时，用一次指数平滑法会出现滞后偏差，此时可以从数据变换的角度来考虑改进措施，即在运用指数平滑法以前先对数据作一些技术上的处理，使之能适合于一次指数平滑模型，以后再对输出结果作技术上的返回处理，使之恢复为原变量的形态。差分方法即为改变数据变动趋势的简易方法。

事实上，MATLAB自带smooth平滑函数，其具体用法见函数说明。

例4.25　某商品2010—2014年销售如表21.9所示。试预测2014年全年的销售量，以便为生产决策。

表 21.9　某商品销售数据

年　份	月　份											
2010	46	66	138	182	384	690	508	244	120	68	38	54
2011	60	74	118	240	622	670	540	246	138	66	47	32
2012	36	40	184	278	648	691	542	384	130	65	38	25
2013	47	65	208	312	752	641	578	323	165	32	76	92
2014	52	70	210	320	740	672	580	340	168	43	74	96

解：

很明显，这是一个具有季节性的时间序列。这里的季节可以是自然季节，也可以是销售生产季节。

对于季节性的时间序列进行完全拟合是非常困难的，但可以通过预测找出季节趋势。

季节性的时间序列预测可以有多种方法，最简单的即为季节系数法，其原理如下。

（1）计算所有数据的平均值T。

（2）计算同季度或同月的数据平均值T_i。

（3）计算季度或月份系数$\beta_i=T/T_i$。

（4）预测：

① 首先计算年份的年加权平均：$y_{m+1}=\dfrac{\sum_{i=1}^{m} w_i y_m}{\sum_{i=1}^{m} w_i}$

其中：w_i为第i年的权重，按自然年份取值；y_m为第i年数据的总和。

② 计算季节或月度平均值：$\overline{y}_{m+1}=\dfrac{y_{m+1}}{n}$，如果为季度则$n$=4，如果为月份则为12；

③ 计算第j个季度或月份的预测值：

$$\hat{y}_{m=+1}=\beta_j\times\overline{y}_{m+1}$$

据此，可编程计算如下。

```
>> x=[46 66 138 182 384 690 508 244 120 68 38 54
      60 74 118 240 622 670 540 246 138 66 47 32
      36 40 184 278 648 691 542 384 130 65 38 25
      47 65 208 312 752 641 578 323 165 32 76 92
      52 70 210 320 740 672 580 340 168 43 74 96];
 >> y=season(x,'m')       %m是指按月份预测
```

例4.26　对表21.10所示的某海洋冰情等级序列进行下一年的冰情预测。

表 21.10　某海洋冰情等级序列实测值　　单位：冰情

年份	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977
等级	3.00	4.50	5.00	3.00	3.50	3.00	1.00	3.00	1.50	1.50	4.50	2.50
年份	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989
等级	2.50	3.00	2.50	2.50	2.00	3.00	3.50	3.00	3.00	2.00	1.50	3.00
年份	1990	1991	1992									
等级	1.50	1.50	1.50									

解：

很明显，这是一个具有季节性的时间序列。这里的季节可以是自然季节，也可以是销售生产季节。

对于季节性的时间序列进行完全拟合是非常困难的，但可以通过预测找出季节趋势。

季节性的时间序列预测可以有多种方法，最简单的即为季节系数法，其原理如下。

（1）计算所有数据的平均值T。

（2）计算同季度或同月的数据平均值T_i。

（3）计算季度或月份系数$\beta_i=T/T_i$。

（4）预测：

① 首先计算年份的年加权平均：$y_{m+1}=\dfrac{\sum_{i=1}^{m} w_i y_m}{\sum_{i=1}^{m} w_i}$

其中：w_i为第i年的权重，按自然年份取值；y_m为第i年数据的总和。

② 计算季节或月度平均值：$\overline{y}_{m+1}=\dfrac{y_{m+1}}{n}$，如果为季度则$n$=4，如果为月份则为12；

③ 计算第j个季度或月份的预测值：

$$\hat{y}_{m=+1}=\beta_j \times \overline{y}_{m+1}$$

据此，可编程计算如下。

```
>> x=[46 66 138 182 384 690 508 244 120 68 38 54
      60 74 118 240 622 670 540 246 138 66 47 32
      36 40 184 278 648 691 542 384 130 65 38 25
      47 65 208 312 752 641 578 323 165 32 76 92
      52 70 210 320 740 672 580 340 168 43 74 96];
>> y=season(x,'m')      %m是指按月份预测
```

例4.26　对表21.10所示的某海洋冰情等级序列进行下一年的冰情预测。

表 21.10　某海洋冰情等级序列实测值　　单位：冰情

年份	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977
等级	3.00	4.50	5.00	3.00	3.50	3.00	1.00	3.00	1.50	1.50	4.50	2.50
年份	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989
等级	2.50	3.00	2.50	2.50	2.00	3.00	3.50	3.00	3.00	2.00	1.50	3.00
年份	1990	1991	1992									
等级	1.50	1.50	1.50									

解：

```
>> x=[3.00  4.50  5.00  3.00  3.50  3.00  1.00  3.00  1.50  1.50  4.50  2.50...
      2.50  3.00  2.50  2.50  2.00   3.00  3.50   3.00  3.00  2.00  1.50  3.00...
      1.50  1.50   1.50 ];
>> m=length(x); plot(1:m,x,'o-')   %画图 21.12
```

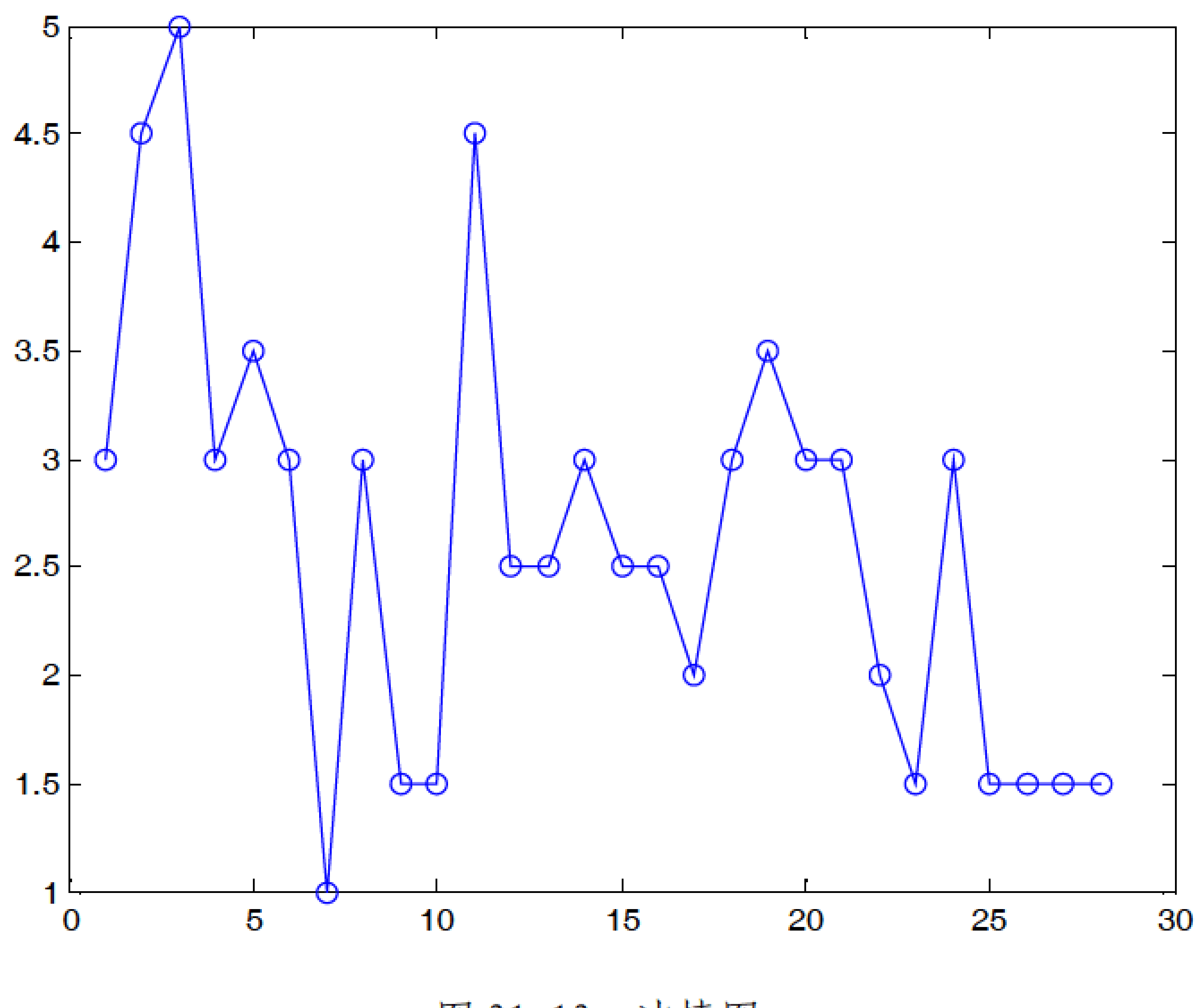


图 21.12　冰情图

可以看出冰情较为复杂。

```
>>y=smoothpre(x,'qE',1);          %二次指数平滑
  y=1.2986                        %实际值为 1.50
>> y=net_time(x);                 %神经网络法
 y=1.5000
```

例 4.27　某县油菜发病率数据为 X_0=(6,20,40,25,40,45,35,21,14,18,15.5,17,15)，试用灰色系统模型进行模拟。

解：

```
>> x=[6 20 40 25 40 45 35 21 14 18 15.5 17 15];
>> [a,b,c,d]=gm(x);
>> a=0.0648    23.3878
```

gm(x)为 GM（1,1）模型函数，其中 *a* 为模型参数，*b* 为各点的模拟值及残差，*c* 为残差平方和，*d* 为平均相对误差。

利用模型参数，便可以进行预测。

例 4.28　某地区平均降水量数据(单位：mm)序列为 X=(390.6 412.0 320.0 559.2 380.8 542.4 553.0 310.0 561.0 300.0 632.0 540.0 406.2 313.8 576.0 587.6 318.5),取 ξ=320mm 为下限异常值(旱灾)，试作旱灾预测。

解：

```
>> x=[390.6  412.0  320.0  559.2  380.8  542.4  553.0  310.0  561.0  300.0  632.0
      540.0  406.2  313.8  576.0  587.6  318.5];
>> [a,b,c]=graynorm(x,320,1)
c=5              %即此序列后的5年后，可能会发生灾变
```

例 4.29　设一随机系统状态空间 $E=\{1,2,3,4\}$，记录观测系统所处状态如下，若该系统可用马氏模型描述，估计转移概率 P_{ij}。

```
4 3 2 1 4 3 1 1 2 3
2 1 2 3 4 4 3 3 1 1
1 3 3 2 1 2 2 2 4 4
2 3 2 3 1 1 2 4 3 1
```

解：

```
>> a=[4 3 2 1 4 3 1 1 2 3 2 1 2 3 4 4 3 3 1 1 1 3 3 2 1 2 2 2 4 4 2 3 2 3 1
      1 2 4 3 1];
>> p=trans_m(a);        %转移矩阵函数
p= 0.4000    0.4000    0.1000    0.1000
   0.2727    0.1818    0.3636    0.1818
   0.3636    0.3636    0.1818    0.0909
        0    0.1429    0.5714    0.2857
```

例 4.30　现在市场上有 A、B、C 三个厂家生产的 50 克袋状味精，用"$\xi n=1$""$\xi n=2$""$\xi n=3$"分别表示"顾客第 n 次购买 A、B、C 厂的味精"。显然，$\{\xi n = 1,2,\cdots,n\}$是一个马氏链。若已知第一次顾客购买三个厂味精的概率依次为 0.2，0.4，0.4。又知道一般顾客购买的倾向由表 21.11 给出。求顾客第 4 次购买各家味精的概率，并预测经过长期的多次购买之后，顾客的购买倾向如何。

表 21.11　顾客购买倾向

		下次购买		
		A	B	C
上次购买	A	0.8	0.1	0.1
	B	0.5	0.1	0.1
	C	0.5	0.3	0.2

解：

```
>> p=[0.8000 0.1000  0.1000;0.5000  0.1000  0.4000; 0.5000  0.3000  0.2000];
>> p1=[0.2 0.4 0.4];
  >> p2=p1*p^3;              %第4次购买不同产品的概率
     p2=0.7004  0.1360  0.1636
  >> y=limit_p(p)            %计算极限概率
```

```
y=0.71430.13100.1548  或  5/7   11/84   13/84
```

例 4.31 表 21.12 为 2002—2006 年某省各地区人均国内生产总值（GDP），请用马尔可夫链方法预测该省 2007 年的发展情况。

表 21.12 GDP 值 单位：元

年份 / 城市	2002	2003	2004	2005	2006
1	21962	25252	29058	39792	44389
2	23570	28024	33544	37457	33734
3	11882	13868	15456	17700	20268
4	8175	8987	9784	11522	13871
5	23412	28825	29536	51811	58051
6	10230	11703	15007	17577	16074
7	19022	22432	29834	39085	49993
8	13031	15488	19917	20645	22478
9	10424	11685	13819	16954	19577
10	12492	15713	17353	17464	19477
11	3711	4052	4229	5820	6813
12	4730	4125	4900	6858	7784
13	7450	8238	8949	10081	9852
14	2350	2544	3039	3878	4490
15	4973	5322	6314	5756	7282
16	5415	6106	7290	8678	10357
17	7153	7977	8989	9360	10644

解：

首先将表中的数据离散化。按一般国际惯例，可以按以下国民生产总值（美元）将发展情况分为：发达（A4）＞3000 $ 、富裕（A3）1500~3000 $ 、小康（A2）800~1500 $ 、温饱（A1）300~800 $ 。

据此，可以将表中分类，得到该省的发展情况，如表 21.13 所示。

表 21.13 发展情况

年份 / 城市	2002	2003	2004	2005	2006
A1	5	5	4	3	1
A2	6	5	4	5	6
A3	6	4	5	5	6
A4	0	3	4	4	4

从而，可以计算出2002—2003年、2003—2004年、2004—2005年和2006—2007年的转移矩阵。

例2002—2003年间的转移矩阵：

```
p1=[1.0000    0         0         0
    0         0.8333    0.1667    0
    0         0         0.5000    0.5000
    0         0         0         1.0000];
```

再求4个转移矩阵的平均值，即为整个的转移矩阵：

```
p=[0.7208    0.2792    0         0
   0         0.8083    0.1917    0
   0         0         0.8125    0.1875
   0         0         0         1.0000];
```

根据表中数据，可知2006年人均GDP的状态数为λ_{2006}=(1,6,6,4)

则2007年的状态为：λ_{2007}=λ_{2006}*p=(0.7208,5.129,6.02,5.125)，实际为λ_{2007}=(1,5,6,5)

即1个城市处于A1状态，5个城市处于A2状态、6个城市处于A3状态、5个城市处于A4状态。

如果求极限概率，可知最终的状态为（0,0,0,1），即各城市都可以达到富裕状态。

例 4.32　有正常骰子和灌铅骰子各一枚，通过实验可得到一系列数字（点数）组成的序列（观察序列），但从序列并不能得知这些数字是用正常骰子还是用灌铅骰子掷出（状态序列），即构成了一个隐马尔可夫链（Hidden Markov Model，HMM）。通过实验可得到如下的转移矩阵（trans）及混淆矩阵(emis)。请分析这个 HMM 模型。

```
trans=[0.9 0.1;0.2 0.8]
emis =[1/6 1/6 1/6 1/6 1/6 1/6;0 1/8 1/8 3/16 3/16 3/8]
```

解：

在MATLAB中的统计工具箱，有专门的有关HMM的函数。利用这些函数可以解决一般的HMM模型。

（1）产生序列：

```
[seq, states]=hmmgenerate(n, trans, emis); %n为序列长度
```

产生观察序列seq及状态序列states。

（2）计算状态序列：

```
likelystates=hmmviterbi(seq, trans, emis);
```

产生与seq相对应的状态序列。可以与实际得到的序列相比较，计算正确率。

（3）对转移矩阵及混淆矩阵作进一步改进（评估问题）：

```
[trans_est, emis_est]=hmmestimate(seq, states);
```

对产生seq、states的转移矩阵和混淆矩阵作进一步改进。

从转移矩阵和混淆矩阵，可以得到两个骰子各点出现的概率。

（4）对初始转移矩阵和混淆矩阵进行学习改进（学习问题）：

```
[trans_est1, emis_est1]=hmmtrain(seq, trans_guess, emis_guess);
```

（5）已知HMM模型及一个观察序列，求状态序列（解码问题）：

```
    pstates=hmmdecode(seq, trans, emis)
```

例 4.33　某地区月平均降水的资料如表 21.14，请对此进行预测分析。

表 21.14　某地区平均降水　mm

降水量	42.4	10.2	116.8	4.8	43.6	13.3	61.6	99.3	139.5	55.5	68.3	83.4	90.0	18.8	47.6	99.6	100.1	80.6
	90.0	100.8	146.1	55.1	172.6	274.8	125.2	4.8	24.2	9.8	19.4	58.6	140.0	38.3	166.8	104.8		

解：

```
>> x=[42.4 10.2 116.8 4.8 43.6 13.3 61.6 99.3 139.5 55.5 68.3 83.4 90.0 18.8 47.6 99.6 100.1...
  80.6 90.0 100.8 146.1 55.1 172.6 274.8 125.2 4.8 24.2 9.8 19.4 58.6 140.0 38.3 166.8 104.8];
>> a=length(x);
>>y=ar(x,5)                          %求五阶 AR 模型
    Discrete-time IDPOLY model: A(q)y(t) = e(t)
     A(q) = 1 - 0.641 q^-1 + 0.03595 q^-2 - 0.1406 q^-3 - 0.1932 q^-4 + 0.06636 q^-5
>> for i=6:a                         %预测值
x2(i)=-(1-0.641*x(i-1)+0.03595*x(i-2)-0.1406*x(i-3)-0.1932*x(i-4)+0.06636*x(i-5));
   end
```

根据计算结果，可作图 21.13，预测结果可以接受。

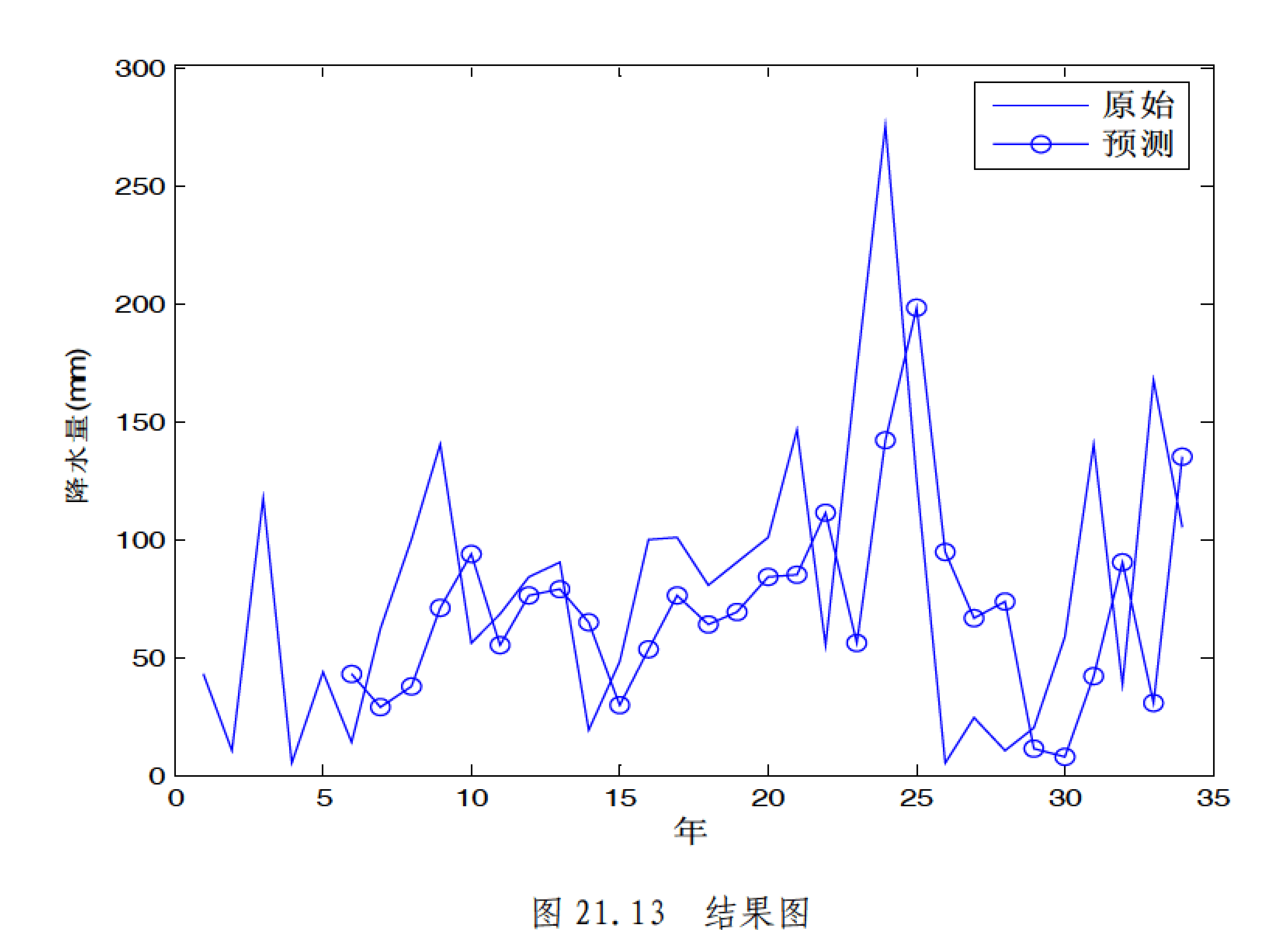


图 21.13　结果图

例 4.34　多个因变量与多个自变量的线性回归问题（简称多对多的线性回归）在实际应用中更为一般和广泛，如生物与环境问题，生物系统中的功能团之间的关系等，均属性此类问题。

表 21.15 为某植物品种区试的资料，其中 x_1 为冬季分蘖，x_2 为株高，y_1 为每穗粒数，y_2 为千粒重，试进行 y_1、y_2 对 x_1、x_2 的回归分析。

表 21.15　品种区试数据

植物品种 \ 性　状	x_1（万）	x_2（cm）	y_1（颗）	y_2（g）
1	11.5	95.3	26.4	39.2
2	9.0	97.7	30.8	46.8
3	7.9	110.7	39.7	39.1
4	9.1	89.0	35.4	35.3
5	11.6	88.0	29.3	37.0
6	13.0	87.7	24.6	44.8
7	11.6	79.7	25.6	43.7
8	10.7	119.3	29.9	38.8
9	11.1	87.7	32.2	35.6

解：

```
>> x=[11.5 95.3; 9.0 97.7;7.9 110.7; 9.1 89.0;11.6 88.0;13.0 87.7…
    11.6 79.7;10.7 119.3;11.1 87.7];
>>y=[26.4 39.2;30.8 46.8;39.7 39.1;35.4 35.3;29.3 37.0;24.6 44.8…
   25.6 43.7;29.9 38.8;32.2 35.6];
>> [beta,stats]=mulregress(x,y); %其中 stats 第 1 列为统计量计算值，第 2 列为查表值
>> beta=58.0806   -2.6490    0.0049              %第 1 个方程式回归系数
        36.9666    0.3472   -0.0065              %第 2 个方程式回归系数
>> stats{1}=[3.8491]    [3.8379]    '回归显著'   %回归式的统计检验
  stats{2}=[14.4020]    [3.8379]    'x1 对 Y 作用不显著'
  stats{3}=[0.0014]     [3.8379]    'x2 对 Y 作用不显著'
  stats{4}=[-4.1089]    [1.9432]    'x1 对 y1 起作用'
  stats{5}=[0.2953]     [1.9432]    'x1 对 y2 不起作用'
  stats{6}=[0.0588]     [1.9432]    'x1 对 y1 不起作用'
  stats{7}=[-0.0431]    [1.9432]    'x1 对 y2 不起作用'
```

例 4.35　已知某一地区 1980—1999 年的肿瘤引起的死亡率（‰）如表 21.16 所示，试建立 AR 模型。

表 21.16　肿瘤死亡率

死亡率	10.010	11.260	9.000	9.090	9.440	9.090	8.730	8.680	9.040	9.045
	10.050	7.330	6.190	5.680	5.860	5.630	5.560	5.640	5.700	6.360

解：

```
>> x=[10.010 11.260 9.000 9.090 9.440 9.090 8.730 8.680 9.040 9.045 10.050 7.330
6.190 5.680 5.860 5.630 5.560 5.640 5.700 6.360];
>> a=lpc(x,3)                                    %计算模型参数
```

```
 a=1.0000   -1.0152   0.1924   -0.1200
>> estx=filter([0-a(2:end)],1,x);          %估计时间序列
>> plot(1:20,x,1:20,estx,'-.');legend('原始信号','LPC 估计'); hold on
>> plot(1:20,x);plot(1:20,x,'o');plot(1:20,estx,'*');xlabel('采样点');ylabel
('幅度')  %图 21.14
```

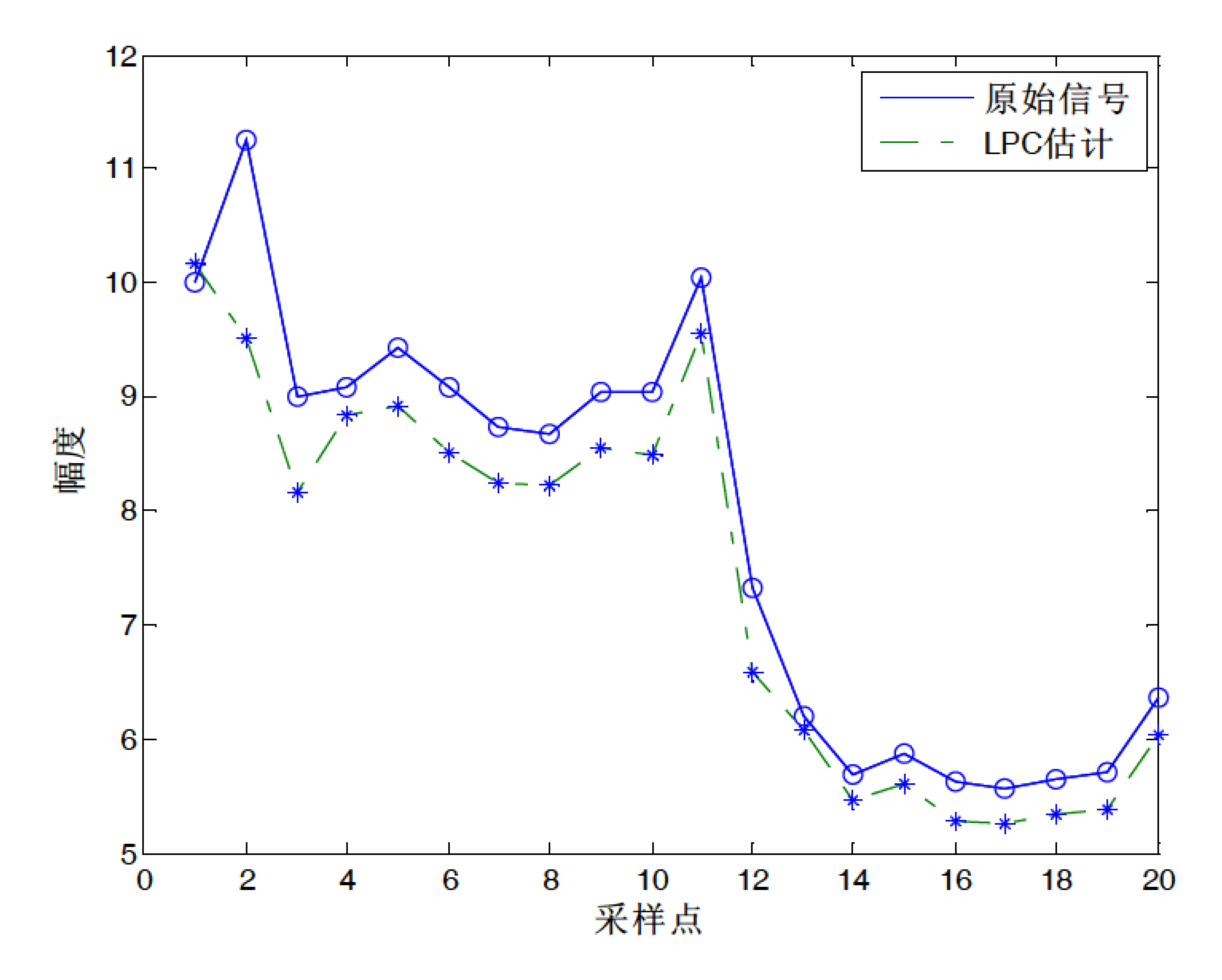


图 21.14　原始信号与估计值

例 4.36　为了提高管理效率，某工厂决定对某工段的用时进行分析。现通过大量的实验得到该工段劳动工时的数据（见 Excel 劳动工时预测数据）。试建立该工时的预测模型。

解：

该数据集是描述某工件的劳动用时，其格式是：

加工宽度	加工直径	加工深度	用工耗时
2	20	1	0.7
2	20	2	0.8
...	...	...	...
2	30	1	0.8

可以看出，三个自变量都为离散型的，回归变量是连续型的，并且实验是按一定的正交表进行的。

自变量为离散型的回归模型有以下几种情况。

（1）自变量全部为离散型，响应变量是连续型，并且试验是按合适的正交表设计量，可以按以下公式计算各参数

$$\widehat{\beta}_0=\frac{T}{n} \qquad \widehat{\beta}_{jk_j}=\frac{r_jT_{(k_j)}^{(j)}}{n}-\frac{T}{n}$$

式中：T 为所有试验结果的和；n 为试验次数；$T_{(k_j)}^{(j)}$ 为第 j 个自变量取水平 k_j 的试验结果和。

否则按下式计算

$$\hat{\beta}=(X'X+L'L)^{-1}y$$

其中 $\boldsymbol{X}$ 为设计矩阵

$$\boldsymbol{X}=\begin{bmatrix}1 & \delta_1(1,1) & \delta_1(1,2) & \cdots\delta_1(m,1)\cdots\delta_1(m,r_m)\\ 1 & \delta_2(1,1) & \delta_2(1,2) & \cdots\delta_2(m,1)\cdots\delta_2(m,r_m)\\ \vdots & \vdots & \vdots & \vdots\\ 1 & \delta_n(1,1) & \delta_n(1,2) & \cdots\delta_1(m,1)\cdots\delta_1(m,r_m)\end{bmatrix}$$

$$\boldsymbol{L}=\begin{bmatrix}0 & I_{r1}^{'} & 0 & 0\\ 0 & 0 & I_{r_2}^{'} & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & I_{r_m}^{'}\end{bmatrix}，I\text{ 为单位矩阵}$$

（2）响应变量是连续型的，回归变量是连续型与离散型混合的。这时要将连续型变量离散化统一变换成离散变量，然后按情况（1）进行处理。

（3）响应变量 y 是离散型的，即 y 只能属于如下 r 个类：$A_1,A_2,\cdots,A_r$。这时将 $y^{(i)}$进行数量化，其方法是：当 $y^{(i)}$ 属于 A_t类时，记为（$y^{(i)}$（A_t））顺序评给一个分数，$y^{(i)}$（A_t）$=t$，t=1,2⋯,r，此时回归预报方程便成为一个判别函数，这样便可以根据 n 次试验 y 所出现的类型得分来确定判别限。

试在 n 次试验中，y 有 n_t次属于 A_t（t=1,2⋯,r）且 $\sum_{t=1}^{r}n_t=n$，用

$$y_t^*=\frac{t*n_t+(t+1)n_{t+1}}{n_t+n_{t+1}},t=1,2,\cdots,r-1$$

作为判别限。若 $\hat{y}^{(i)}<y_1^*$ 则认为 $y^{(i)}$属于 A_1 类；若 $y_{t-1}^*\leqslant\hat{y}^{(i)}<y_t^*,t=2,3,\cdots,r-1,$ 则认为 $y^{(i)}$属于 A_t 类；若 $\hat{y}^{(i)}\geqslant y_{r-2}^*$ 则认为 $y^{(i)}$属于 A_r 类。

据此，可编程计算如下。

```
>>x=xlsread('D:\劳动工时预测数据',1,'A2:D237');       %读入数据
>> [beta,resid,R]=discrete_regress(x,1);
>> [beta,resid,R]=discrete_regress(x,2);
```

比较两种方法计算的结果，可以看出第 2 种方法得到的回归系数为 0.9748，大于第 1 种方法所得到的 0.7423，第 1 种方法所得到的残差也小得多。这说明实验有可能不是完全按照正交表所设计的。

第 22 章

聚　类

22.1 聚类分析概述

聚类是将数据集划分为若干相似对象组成的多个组（group）或簇（cluster）的过程，使得同一组中对象的相似度最大化，不同组中对象间的相似度最小化。相似度可以根据描述对象的属性值计算，对象间的距离是最常采用的度量指标。

聚类分析是数据分析中的一种重要技术，它的应用极为广泛。许多领域中都会涉及聚类分析方法的应用与研究工作。商业上聚类分析是细分市场的有效工具，基于消费者行为来发现不同类型的客户群，并刻画不同客户群的特征；在保险行业中，聚类分析通过消费特征来鉴定汽车保险单持有者的分组；在房地产行业中，聚类分析根据住宅类型、价值和地理位置等特征来鉴定一个城市的房产分组。

从统计学的观点看，聚类分析是通过数据建模简化数据的一种方法。作为多元统计分析的主要分支之一，聚类分析方法包括系统聚类法、加入法、动态聚类法、有序样品聚类等，主要的度量是距离或相似度。

从机器学习的角度来看，簇相当于隐藏模式。聚类是搜索簇的无监督学习。与分类不同，无监督学习不依赖于预先定义的类或带类标号的训练实例，它是以某种距离度量为基础，将所有对象进行聚类，使得同一聚类间的距离最小，不同聚类之间的距离最大。聚类分析可以在几乎没有相关数据先验信息（如统计模型）可用的情况下分析数据点中的内在关系以进行进一步分析。

从实际应用的角度看，聚类分析是数据挖掘的主要任务之一。例如，在科学数据探测、信息检索、文本挖掘、Web 数据分析等方面的数据挖掘中，聚类分析技术都起着重要作用。在商业领域，聚类可以帮助市场经营人员分析客户数据库、发现不同类型的客户群，按购买习惯分类并描述客户群的特征。在生物学界，聚类可以用于动物和植物分类，对具有相似功能的基团进行分类，了解种群的内存结构。

就数据挖掘功能而言，聚类能够作为一个独立的工具获得数据的分布状况，了解各数据组的特征，确定所感兴趣的数据组以做进一步的分析，还可以作为其他数据挖掘任务（或分类、关联分析）的预处理步骤，以在聚类分析所生成的簇上进一步处理。

聚类分析是一个具有很强挑战性的领域，它的一些潜在应用对分析算法提出了特别的要求，下面列出一些典型的要求。

（1）处理不同类型属性的能力。在实际应用中，聚类算法不仅能用于数据值类型数据，而且也要适应于其他类型的数据结构，如二元类型、分类（标称）类型、序数类型、混合类型等。

（2）对大型数据集的可扩展性。许多聚类算法在小数据集上有效。随着大型数据库、数据仓库的广泛应用，对大数据集聚类时许多原有的聚类算法可以产生偏差，甚至出现错误的结果。因此需要研究具有良好可扩展性的聚类算法。

（3）处理高维数据的能力。大型数据库或数据仓库可能有若干个维或属性，因此聚类算法要有处理高维数据的能力，尤其当数据稀疏、高度倾斜时更是如此。

（4）发现任意形状簇的能力。许多聚类算法是建立在距离度量基础上的，倾向于生成球形的、大小和密度相近的簇。但是，数据集中实际存在的簇可能是任意形状，簇的大小差异较大，密度也不尽相同。研究能够发现任意形状簇的聚类算法是非常必要的。

（5）处理孤立点或“噪声”数据的能力。数据集合中往往包含孤立点、缺失值、未知或错

误的数据。处理孤立点时，应该考虑两个方面：①某些实际问题可以要求聚类算法对“噪声”数据具有较低的敏感度，以免导致低质量的聚类结果，因此，算法应考虑排除或降低来自孤立点的影响；②一些实际问题（如对商业欺诈的分析）要求聚类算法合理地发现孤立点，而不是如①中的聚类算法那样将孤立点排除掉或尽量减少来自孤立点的影响。孤立点探测和分析是一个有实际意义的数据挖掘任务，称为孤立点挖掘。

（6）对数据顺序的不敏感性。为了提高聚类结果的稳定性，应该研究对输入数据顺序不敏感的聚类算法。

（7）对先验知识和用户自定义参数的依赖性。许多聚类算法要求输入特定的参数，如产生的簇的数目。一方面参数很难确定，尤其是对高维数据集；另一方面，这类算法往往对输入参数具有敏感性，参数的细微变化可以导致显著不同的聚类结果，另外参数设置加重了用户负担，也难以控制聚类结果的质量。

（8）聚类结果的可解释性和实用性。聚类结果应该是可理解的、可解释的和可用的。

（9）基于约束的聚类。现实应用中总会出现其他约束条件，聚类算法在考虑这些限制的情况下，仍有较好的聚类结果。

聚类算法主要可以分为划分法、层次法、基于密度的方法、基于网络的方法和基于模型的方法。要注意的是部分聚类算法如支持向量机只能处理二分类问题。对于多分类问题一般是将其转化为多个二分类问题，即将数据集分成为多个二分类问题的数据子集，在每一个子集上进行聚类分析，输出为各个分类器结果的组合。也可以每次只使用某两类的数据进行聚类分析，对于一个未知的测试数据，其输出是支持度最高的那个类。

21.2 聚类分析中的数据类型

相似性度量是衡量变量间相互关系强弱、联系紧密程度的重要方法，是聚类分析的基础，其方式与数据类型密切相关，数据类型不同，相似性度量的方式也不同。

从广义上讲，数据是记录在介质上的信息（在不同的场合可以称为数据对象、点、向量、模式、事件、案例、样本或实体等），它是数据及其属性的集体，其表现形式可以是数字、符号、文字、图像或计算机代码等。

对于数据的理解不仅需要了解其表现形式，而且还需要了解数据的语义，即对数据含义的说明，它是数据对象（记录）所有属性的集合。

属性（也称为特征、维或字段）是指一个对象的某方面性质或特征。一个对象通过若干个属性来刻画其特征。根据其属性的不同，属性可分类标称（Nominal）、序数（Ordinal）、区间（Interval）和比率（Ratio）。

（1）标称属性，其值提供足够的信息以区分对象，如颜色、性别、产品编号等。这种属性值大小的比较没有实际意义。

（2）序数属性，其值提供足够的信息以区分对象，如客户等级（贵客卡、银卡、金卡、钻石卡）、企业信用评估等级、奖励等级等。

（3）区间属性，其值之间的差是有意义的，但比率及和是没有意义的。如开户日期、摄氏温度。

（4）比率属性，其值之间的差和比率都是有意义的，如年度消费总额、通话时长等。

误的数据。处理孤立点时，应该考虑两个方面：①某些实际问题可以要求聚类算法对“噪声”数据具有较低的敏感度，以免导致低质量的聚类结果，因此，算法应考虑排除或降低来自孤立点的影响；②一些实际问题（如对商业欺诈的分析）要求聚类算法合理地发现孤立点，而不是如①中的聚类算法那样将孤立点排除掉或尽量减少来自孤立点的影响。孤立点探测和分析是一个有实际意义的数据挖掘任务，称为孤立点挖掘。

（6）对数据顺序的不敏感性。为了提高聚类结果的稳定性，应该研究对输入数据顺序不敏感的聚类算法。

（7）对先验知识和用户自定义参数的依赖性。许多聚类算法要求输入特定的参数，如产生的簇的数目。一方面参数很难确定，尤其是对高维数据集；另一方面，这类算法往往对输入参数具有敏感性，参数的细微变化可以导致显著不同的聚类结果，另外参数设置加重了用户负担，也难以控制聚类结果的质量。

（8）聚类结果的可解释性和实用性。聚类结果应该是可理解的、可解释的和可用的。

（9）基于约束的聚类。现实应用中总会出现其他约束条件，聚类算法在考虑这些限制的情况下，仍有较好的聚类结果。

聚类算法主要可以分为划分法、层次法、基于密度的方法、基于网络的方法和基于模型的方法。要注意的是部分聚类算法如支持向量机只能处理二分类问题。对于多分类问题一般是将其转化为多个二分类问题，即将数据集分成为多个二分类问题的数据子集，在每一个子集上进行聚类分析，输出为各个分类器结果的组合。也可以每次只使用某两类的数据进行聚类分析，对于一个未知的测试数据，其输出是支持度最高的那个类。

21.2　聚类分析中的数据类型

相似性度量是衡量变量间相互关系强弱、联系紧密程度的重要方法，是聚类分析的基础，其方式与数据类型密切相关，数据类型不同，相似性度量的方式也不同。

从广义上讲，数据是记录在介质上的信息（在不同的场合可以称为数据对象、点、向量、模式、事件、案例、样本或实体等），它是数据及其属性的集体，其表现形式可以是数字、符号、文字、图像或计算机代码等。

对于数据的理解不仅需要了解其表现形式，而且还需要了解数据的语义，即对数据含义的说明，它是数据对象（记录）所有属性的集合。

属性（也称为特征、维或字段）是指一个对象的某方面性质或特征。一个对象通过若干个属性来刻画其特征。根据其属性的不同，属性可分类标称（Nominal）、序数（Ordinal）、区间（Interval）和比率（Ratio）。

（1）标称属性，其值提供足够的信息以区分对象，如颜色、性别、产品编号等。这种属性值大小的比较没有实际意义。

（2）序数属性，其值提供足够的信息以区分对象，如客户等级（贵客卡、银卡、金卡、钻石卡）、企业信用评估等级、奖励等级等。

（3）区间属性，其值之间的差是有意义的，但比率及和是没有意义的。如开户日期、摄氏温度。

（4）比率属性，其值之间的差和比率都是有意义的，如年度消费总额、通话时长等。

属性可以进一步归类为两种。

（1）分类或定性属性，包括标称和序数属性，取值为集合。

（2）数据或定量属性，包括区间和比率属性，取值为区间，可以是整数值或连续值。

通常数据挖掘算法以表格形式组织数据以形成数据集，如表 22.1 所示。但也有可能是其他的形式，需要经过适当的预处理。

表 22.1　数据集格式样本

客户编号	客户类型	行业大类	通话级别	通话总费用（元）
N22011002518	大客户	学校	市话＋国内长途	16326
C14005889674	商业客户	一般制造业	市话＋国内长途	27594
N22005673821	商业客户	批发和零售业	市话＋国内长途	63748
3253789	大客户	房地产和建筑业	市话＋国际长途＋国内长途	80384
DI400982435	大客户	银行	市话＋国际长途	59873

数据集需要考虑三个方面的问题。

（1）维度：它是指数据集中的对象具有的属性个数总和。根据维度的大小可以将数据集分为高、中、低维数据集。维度越高，计算越复杂，经常会遇到“维灾难”的情况，所以在数据挖掘中一般需要对高维数据进行降维处理。

（2）稀疏性：它是指数据集中有意义的数据非常少。超市购物记录、文本数据集具有典型的稀疏性。数据的稀疏性影响数据的有效性、存储方式等方面。

（3）分辨率：可以在不同的分辨率或粒度下观察数据，而且在不同的分辨率下对象的性质也不同。数据中隐藏的模式依赖于分辨率，分辨率太高、太低，都得不到有效的模式，针对具体应用，需要选择合适的分辨率或粒度。

数据集的类型可以分成以下三类。

1. 记录数据

一般的数据挖掘任务都是假定数据集是记录（数据对象）的集合，每个记录都由相等数目的属性构成，记录之间或属性之间没有明显的联系。记录数据通常存放在平面文件或关系数据库中。根据数据挖掘任务的不同要求，记录数据也可以有不同的种类。

（1）事务数据或购物篮数据。

事务数据是一种特殊类型的记录数据，其中每个记录涉及一个项的集合。典型的事务数据如超市零售数据，顾客一次购物所购买的商品的集合就构成一个事务，而购买的商品就是项。这种类型的数据也称为购物篮数据。

（2）数据矩阵。

如果一个数据集中的所有数据对象都具有相同的数据属性集，由该数据对象可以看作多维空间中的点（向量），其中每一维代表描述对象的不同。这样的数据对象集可以用一个 $n\times m$ 的矩阵来表示，其中 n 为对象数（行或列），m 为属性数（列或行）。数据矩阵是记录数据的变体，可以使用标准的矩阵操作对数据进行变换和操纵。因此，对于大部分统计数据，数据矩阵是一种标准的数据格式。

文本数据是数据矩阵的一种特殊情况，可以用稀疏矩阵表示，其中属性类型相同并且是非对称的，即只有零值才是重要的。在信息检索领域，文本被看成是出现在文本中的关键词的集合，这些关键词就是特征项。利用特征项，文本可以表示成布尔模型、向量模型和概率模型。特别地，如果忽略文档中词的次序，则文档可以用词向量表示，其中每个词是向量的一个分量（属性），而每个分量的值对应词在文档中出现的次数。

2. 基于图形的数据

有时，图形可以方便而有效地表示对象之间的关系。

（1）带有对象之间联系的数据：对象之间的联系常常携带重要的信息。在这种情况下，数据常用图形表示。特殊地，数据对象映射到图的特点，而对象之间的联系用对象之间的链、方向、权值等表示。例如，万维网的网页上包含文本和指向其他页面的链接，电话通信中形成不同的社会网络群。

（2）具有图形对象的数据：如果对象具有结构，即对象包含具有联系的子对象，则这样的对象常用图表示。例如化合物的结构可以用图形表示，其中节点是原子，节点之间的链是化学键。

3. 有序数据

对于某些数据类型，属性具有涉及时间或空间序的联系。

（1）时序数据或时态数据，可以看作记录数据的扩充，其中每个记录包含一个与之相关联的时间，通常存放包含时间相关属性的关系数据。这些数据可能涉及若干时间标签，每个都具有不同的意义。例如，在超市的数据库中，可以从时间数据上分析出某商品的消费季节，每位顾客的消费周期及偏好。

（2）序列数据是一个数据集合，是个体项的序列，如词或字母的序列，用来存放具有不同或不具有具体时间概念的有序事件的序列，或者顾客购物序列、Web 点击流和生物学序列等。

（3）时间序列数据是一种特殊的时序数据，其中每个记录都是一个时间序列，即一段时间的测量序列，如股票交易、库存挖掘和自然现象等。在分析时间序列数据时，重要的是考虑时间自相关，即如果两个测量的时间很接近，则这些测量的值通常非常相似。

（4）空间数据包含涉及空间的数据，或地理信息系统、医学图像等。空间数据的一个重要特点是空间自相关性，即物理上靠近的对象在其他方面也相似，如地球上相互靠近的两个点通常具有相近的气温和降水量。

（5）流数据是一种可以动态地从观测台流进和流出的数据，具有海量甚至是无限的，动态变化的，以固定的次序流进和流出，只允许一遍或几遍扫描，要求快速响应等特点。数据流的典型例子包括电力供应、网络通信、股票交易、银行、电信及气象等行业数据。

22.3　相似性度量

通常具有若干属性的对象间的相似性用单个属性的相似性的组合来定义。

22.3.1 属性间的相似性度量

1. 标称和区间属性

由于标称属性只含有对象的相异性信息，因此两个对象只有相同或不同的值，如果属性值匹配，则相似度定义为 1，否则为 0；相异度则与之相反，即属性值匹配，相异度为 0，否则为 1。

对于区间属性，则用它们的差值的绝对值来度量相异性。表 22.2 为不同属性情况下的属性相似度度量方法。

表 22.2 简单属性的相异度与相似度定义

属性类型	相异度	相似度
标称型	$d=\begin{cases}0 & 如果x=y\\1 & 如果x\neq y\end{cases}$	$s=1-d=\begin{cases}1 & 如果x=y\\0 & 如果x\neq y\end{cases}$
区间	$d=\|x-y\|$	$s=\dfrac{1}{1+d},s=e^{-d},s=1-\dfrac{d-\min_d}{\max_d-\max_d}$

2. 序数和比率数值属性

（1）序数属性。序数属性变量包括分类的和连续的两种类型。一个分类序数属性与一个标称属性类似，不同的是对应 M 个状态的 M 个顺序值是按一定次序排列的，它有助于记录一些不便于客观度量的主观评价。例如职称就是一个分类的序数属性，按助教、讲师、副教授、教授的顺序排列。一个连续的序数属性看上去就像一组未知范围的连续数据，但它的相对位置要比它的实际数值有意义，顺序是主要的，实际大小是次要的。如比赛的名次，通常名次比排名的具体位置更有意义。一个序数属性的集合可以映射到一个等级集合上，然后通过等级来描述差异，其差异程度计算如下。在以下定义中，假设区间是相等的，但事实可能并非如此。

- 属性 f 有 M_f 个有序状态，将属性值 x_f 替换为相应的等级 r_f， $r_f\in(1,2,\cdots,M_f)$
- 对序数属性等级 r_f 作变换 $z_f=\dfrac{r_f-1}{M_f-1}$，将其映射到区间[0,1]上。
- 利用有关区间属性的任一种距离计算公式来计算差异程度。

例如考虑一个在标度｛poor、fair、ok、good、woonderful｝测量糖的质量的属性，产品 1、2 和 3 分别评定为 woonderful、good 与 ok。为了评价产品的相似度，可以将属性映射到某一等级上｛poor = 1 、fair = 2 、ok = 3 、good = 4 、woonderful = 5｝，这样就可以计算产品间的相异度：$d(1,2)=(5-4)/4=0.25$，$d(1,3)=(5-3)/4=0.5$。可见，产品 1 与 2 较为接近，符合直观观察。

（2）比率数值属性。比率数值属性是在非线性尺度上取得的测量值，例如指数比率，可以近似描述为，典型的例子有细胞繁殖增长的数目描述。在计算这类数值变量所描述对象间的距离时，有以下三种方式。

- 将比率数值变量当作区间间隔数量数值变量来进行计算处理。该方法可能会导致非线性的比例尺度扭曲。
- 将比率数值变量看成是连续的序数属性进行处理。
- 根据实际情况，利用一定的变换方式（如对数变换 $y_f=\log(x_f)$）来处理得到的新变量 y_f

并将其当作区间变量进行处理。此方法效果较好。

22.3.2　对象间的相似性度量

对象间的相似性度量即为多个属性整体的相似性度量的计算，它涉及描述对象的属性类型，需要将不同属性上的相似度整合成一个总的相似度来表示。

假设使用个属性来描述数据记录，将每条记录看成维空间中的一个点，相互间距离越小，相似系数越大的记录间的相似程度越大。

22.3.3　相异度矩阵

按 n 个对象两两间的相异度构建 n 阶矩阵，它是对称的，只需写出上三角或下三角即可。

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ \vdots & \vdots & \vdots & \\ d(n,1) & d(n,2) & \cdots & \cdots 0 \end{bmatrix}$$

其中：$d(i,j)$表示对象 i 与 j 的相异度，它是一个非负的数值。当对象 i 和 j 越相似或“接近”时，$d(i,j)$值越接近于 0；而对象 i 和 j 越不相同或相距“越远”时，$d(i,j)$值越大。相异度矩阵是对象—对象结构的一种数据表达方式。

多数聚类算法都建立在相异度矩阵基础上，如果数据是以数据矩阵形式给出的，就要将数据矩阵转化为相异度矩阵。

计算对象间距离是经常采用的求相异度方法。设两个 p 维向量 $\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{ip})^{\mathrm{T}}$ 和 $\boldsymbol{X}_j=(x_{j1},x_{j2},\cdots,x_{jp})^{\mathrm{T}}$ 分别表示两个对象，根据属性特征的不同，有多种形式的距离度量可以采用。

1. 数值属性相似性度量

（1）距离度量。

① 闵可夫（Minkowski）距离。

对于任意样本对象 $p=[p_1,p_2,\cdots,p_m]$与 $q=[q_1,q_2,\cdots,q_m]$，它们之间闵可夫（Minkowski）距离定义为

$$d_x(p,q)=\left[\sum_{i=1}^{m}|p_i-q_j|^x\right]^{\frac{1}{x}}$$

其中：$x\in[1,\infty]$。

闵可夫距离是无限个距离度量的概化，当 x=1 时为曼哈坦（Manhattan）距离：$d_1(p,q)=\sum_{i=1}^{m}|p_i-q_j|$，当 x=2 时为欧几里得（Euclidean）距离：$d_2(p,q)=\sqrt{\sum_{i=1}^{m}|p_i-q_j|^2}$，当 $x\to\infty$时为切比雪夫（Chebyshev）距离：$d_\infty(p,q)=\max\limits_{1\leqslant i\leqslant m}|p_i-q_j|$。

令对象的维数 p=2，在二维空间中考虑到原点为常数的所有点形成的形状，可以直观地看出：菱形对应于曼哈坦距离；圆形对应于欧几里得距离；方形对应于切比雪夫距离，如图 22.1 所示。

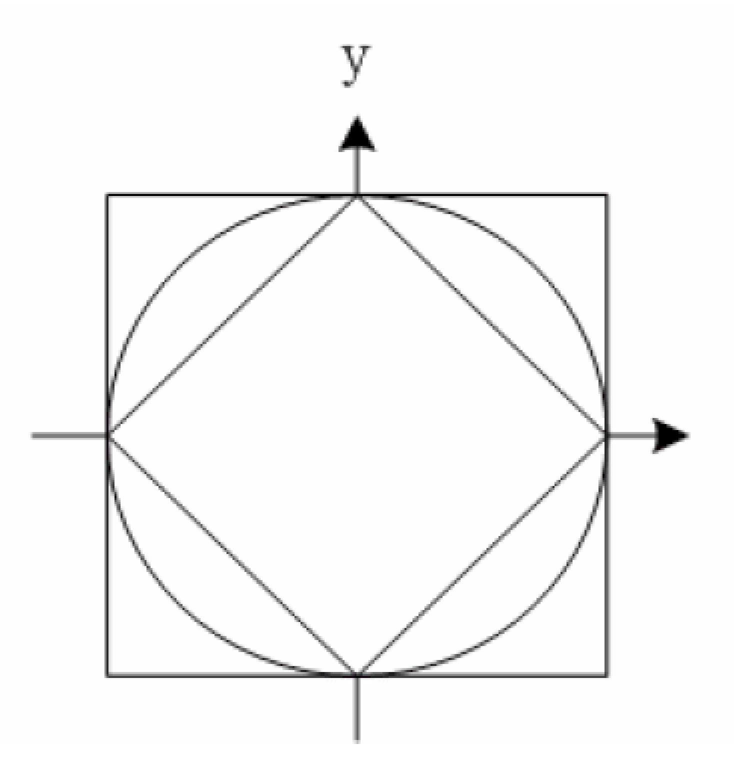


图 22.1　几种距离下与原点相距为常数的点形成的形状

直接使用闵可夫距离的缺点是量纲或度量单位对聚类结果有影响，为消除此影响，通常需要对数据进行规范化。

② Canberra 距离。

$$d_{canb}(p,q)=\sum_{i=1}^{m}\frac{|p_i-q_i|}{|p_i|+|q_i|}$$

Canberra 距离可以看成一个相对 Manhattan 距离，它克服了 Minkowski 距离受量纲影响的缺点。Canberra 距离对默认值是稳健的，当两个坐标都接近于 0 时，Canberra 距离对微小的变化很敏感。

③马哈拉诺比斯距离。

$$d_A(X_i,X_j)=(X_i-X_j)^T A(X_i-X_j)$$

其中：$\boldsymbol{A}$ 为正定矩阵。

在以上距离度量表达式中，还可以根据每个变量的重要性为其赋一个权重，如加权的欧几里得距离形式为

$$d_2(X_i,X_j)=\left\|X_i-X_j\right\|_2=\left[\sum_{k=1}^{p}w_k\,|x_{ik}-x_{jk}|^2\right]^{1/2}$$

（2）相似系数。

距离度量还可以利用基于相似系数定义的距离，它多用于变量指标的相似性度量。

两个对象间的相似系数可以有多种定义形式，常用的有以下几种。

① 夹角余弦。

$$\cos(p,q)=\frac{\sum_i p_i\times q_i}{\sqrt{(\sum_i p_i^2)\times(\sum_i q_i^2)}}$$

夹角余弦忽略各个向量的绝对长度，着重从形状方面考虑它们间的关系。取值范围在[−1,1]。

② 相关系数。

$$\mathrm{Corr}(p,q)=\frac{\sum_i (p_i-\overline{p})\times(q_i-\overline{q})}{\sqrt{(\sum_i (p_i-\overline{p})^2\times\sum_i (q_i-\overline{q})^2)}}$$

其中：$\overline{p}$、$\overline{q}$ 为均值。

相关系数是向量标准化后的夹角余弦，取值范围在区间[-1,1]，它表示两个向量的线性相关程度。

③ 广义 Jaccard 系数。

广义 Jaccard 系数又称 Tanimoto 系数，取值范围在区间[0,1]之间，广泛用于信息检索与生物学分类中，在二元属性情况下简化为系数。

$$\mathrm{EJ}(p,q)=\frac{\sum_i p_i \times q_i}{\sum_i p_i^2+\sum_i q_i^2-\sum_i p_i \times q_i}$$

2. 二值属性的相似性度量

一个二值属性变量只有 0 或 1 两种状态，表示属性的存在与否。假设二值属性对象 p 和 q 取值情况如表 22.3 所示，其中 n_{11} 表示对象 p 和 q 中均取值 1 的二值属性个数，n_{10} 表示对象 p 取 1 而对象 q 取 0 的二值属性个数，n_{01} 表示对象 p 取 0 而对象 q 取 1 的二值属性个数，n_{00} 表示对象 p 和 q 均取 0 的二值属性个数。

表 22.3　二值属性对象 p 和 q 的取值情况

对象 p 取值 / 对象 q 取值	1	0	合　计
1	n_{11}	n_{10}	$n_{11}+n_{10}$
0	n_{01}	n_{00}	$n_{01}+n_{00}$
合　计	$n_{11}+n_{01}$	$n_{10}+n_{00}$	

二值属性相似性存在对称和不对称两种情况。如果一个二值属性的两种状态所表示的内容同等重要，则它是对称的，否则为不对称的。例如，给定属性变量 smoker，它描述一个病人是否吸烟的情况，用 0 或 1 进行编码来表示一个病人吸烟状态是同等重要的，因此是对称变量。基于对称二值变量所计算的相似度称为不变相似性（即变量编码的改变不会影响计算结果）。对于不变相似性，常用简单匹配相关系数来描述对象 p 和 q 之间的差异程度，其定义为

$$d(p,q)=\frac{n_{01}+n_{10}}{n_{00}+n_{01}+n_{10}+n_{11}}$$

其中：$n_{10}+n_{01}$ 表示取值不同的属性个数；$n_{00}+n_{11}$ 表示取值相同的属性个数。

对于不对称的二值变量，如果认为取值 1 比取值 0 更重要、更有意义，那么这样的二值变量就好像只有一种状态。例如属性 disease 的检测结果是阳性（P）或阴性（N），显然这两个检测结果的重要性是不一样的。通常将少见而重要的情况用 1 表示（例 HIV 阳性），而将其他不重要的情况用 0 表示（例如 HIV 阴性），这种情况下对象 p 和 q 之间的差异程度评价通常采用 Jaccard 系数，其定义为

$$d(p,q)=\frac{n_{01}+n_{10}}{n_{01}+n_{10}+n_{11}}$$

3. 混合属性相似性度量

在实际应用中，数据对象往往包含多种类型的属性，因此使用混合类型的属性描述。这需

要将不同类型的属性差异度组合成一个整体，把所有属性间的差异转换到区间[0,1]中。

假设数据集包含 m 个不同类型的属性，对象 p 和 q 之间的差异度距离（推广闵可夫距离）定义为

$$d_x(p,q)=\left(\frac{\sum_{f=1}^{m}\delta_{pq}^{(f)}d_f(p,q)^x}{\sum_{f=1}^{m}\delta_{pq}^{(f)}}\right)^{\frac{1}{x}}$$

其中，如果 p_f或 q_f数据不存在（对象 p 或对象 p 的属性 f 无测量值），或 $p_f=q_f=0$，且属性 f 为非对称二值属性，则标记 $\delta_{pq}^{(f)}=0$，否则 $\delta_{pq}^{(f)}=1$，$\delta_{pq}^{(f)}$ 表示属性 f 为对象 p 和对象 q 之间的差异程度所做的贡献，对象 p 和对象 q 在属性上的相异度 $d_f(p,q)$ 根据其属性类型不同进行相应计算。

- 若属性 f 为二元属性或标称属性，则：如果 $p_f=q_f$，那么 $d_f(p,q)=0$，否则 $d_f(p,q)=1$。
- 若属性 f 为序数型属性，计算对象 p 和对象 q 在属性 f 上的秩（或等级）r_{pf}和 r_{qf}，$d_f(p,q)=\frac{|r_{pf}-r_{qf}|}{M_f-1}$。
- 若属性 f 为区间标度属性，则 $d_f(p,q)=\frac{|p_f-q_f|}{\max_f-\min_f}$，$\max_f$、$\min_f$分别表示属性 f 的最大值和最小值。
- 若属性 f 为比率数值属性，则可以通过变换将其转换成区间标度属性来处理。

这样，当描述对象的属性是不同类型时，对象之间的相异度也能够计算，且取值在[0,1]区间。

4. 由距离度量转换而来的相似性度量

可以通过一个单调递减函数，将距离转换成相似性度量。相似性度量的取值一般在区间[0,1]之间。值越大，说明两个对象越相似。常用的方式有：

- 采用负指数函数将距离转换为相似性度量 s，即：$s(p,q)=e^{-d(p,q)}$
- 采用距离的倒数作为相似性度量，即：$s(p,q)=\frac{1}{1+d(p,q)}$
- 若距离在 0～1 之间，可采用与 1 的差作为相似系数，即：$s(p,q)=1-d(p,q)$

在聚类分析中需要根据数据类型、应用目标等因素选择距离函数。

22.4 聚类的特征

聚类是相似事物的集合，从数学角度则难以给出一种通用严格的定义，常用的有以下几种定义形式，可以适用于不同的场合。

设 G 为元素的集合，它共有 m 个元素，记为 g_i，$i=1,2,\cdots,m$，另外给定一个阈值 $T>0$，则有以下几种类定义：

（1）若 G 中任意两个元素 g_i 和 g_j 之间的距离不大于阈值，即有 $d_{ij}\leq T$，则称 G 为类。

（2）若 G 中任意元素 g_i 与其他元素间的距离均值不大于阈值，即有 $\frac{1}{k-1}\sum_{1\leqslant j\leqslant k} d_{ij} \leqslant T$，则称 G 为类。

（3）对 G 中任意元素 g_i，总存在另一个元素 g_j，它们的距离不大于阈值，即有 $d_{ij}\leqslant T$，则称 G 为类。

若将 G 的元素 g_i 视为随机向量 x_i，则可用以下几种特征来描述类。

（1）类的重心。

类的重心即为各元素均向量

$$x_G = \frac{1}{m}\sum_{i=1}^{m} x_i$$

（2）类的样本离差矩阵与样本协方差矩阵。

它们的定义分别为

$$A_G = \sum_{i=1}^{m}(x_i - x_G)(x_i - x_G)^T$$

$$S_G = \frac{1}{m}A_G$$

（3）类的直径。

类的直径也有多种定义，比较简单的有

$$A_G = \sum_{i=1}^{m}(x_i - x_G)(x_i - x_G)^T$$

22.5　聚类准则

在模式分类中，可以有多种不同的聚类方式，将未知类别的样本划分到对应的类中。在这个过程中，需要确定一种聚类准则来评价各种聚类方法的优劣。事实上各种聚类方法的优劣只是就某种评价准则而言，任何一种聚类方法要满足各种聚类准则是非常困难的。

聚类准则的确定主要有两种方式。

（1）试探方式。凭直觉和经验，针对实际问题给定一种模式相似性测度的阈值，按最近邻规则指定待分类样本属于某一类。例如在以“距离”为相似性测度时，规定一个阈值，如果待测样本与某一类的距离小于阈值，则归入该类。

（2）聚类准则函数法。定义一种聚类准则函数，其函数值与样本的划分有关，当此值达到极值时，就认为样本得到了最佳的划分。常用的聚类函数有误差平方和准则及类间距离和准则。

① 误差平方和准则。

误差平方和也称为类内距离和准则，是一种简单而又应用广泛的聚类准则，其表达式为

$$J = \sum_{i=1}^{m}\sum_{X^j\in\omega_i} \| X - \mu_i \|^2$$

式中：μ_i 为类 ω_i 的均值；J 为样本与聚类中心的函数，表示各样本到其被划并类别的中心的距离之平方和。最佳的划分就是使 J 最小的那种划分。

该准则适用同类样本比较密集，各类样本数目相差不大，而且类间距离较大时的情况。当各类样本数相差很大且类间距离较小时，采用该准则就有可能将样本数多的类拆成两类或多类，从而出现错误聚类。

② 类间距离和准则或离散度准则。

类间距离和定义为

$$J=\sum_{i=1}^{m}(\mu_i-\mu)^T(\mu_i-\mu)$$

其中：μ_i、μ 分别为类 ω_i 和全部样本的均值。

加权的类间距离和定义为

$$J=\sum_{i=1}^{m}\frac{N_i}{N}(\mu_i-\mu)^T(\mu_i-\mu)$$

对应一种划分，可求得一个类间距离和。类间距离和准则是找到使类间距离和最大的那种划分。

事实上，类间距离的和类内距离的统称为离散度矩阵。

类内离散度矩阵 S_i 和总类内离散度矩阵 S_w 分别为

$$S_i=\sum_{x\in\omega_i}(x-\mu_i)(x-\mu_i)^T$$

$$S_w=\sum_{i=1}^{c}S_i$$

类间离散度矩阵

$$S_B=\sum_{i=1}^{c}n_i(\mu_i-\mu)(\mu_i-\mu)^T$$

总离散度矩阵

$$S_T=\sum_{x\in X}(x-\mu)(x-\mu)^T$$

如果采用最小化类内离散度矩阵的迹作为准则函数，可以同时最小化类内离散度迹和最大化类间离散度离散度迹。

22.6 划分方法

对于一个给定的 n 个对象或元组的数据库，采用目标函数最小化的策略，通过迭代把数据分为 k 个块，每个块为一个簇，这就是划分方法。划分方法要满足两个条件：一是每个分组至少包含一个对象；二是每个对象必属于且仅属于某一个分组。

常见的划分方法有 k-均值方法和 k-中心点方法。其他方法都是这两种方法的变形。

k-均值方法和 k-中心点方法已在第 2 篇中做过介绍，在此只介绍 EM 算法。

EM 算法即为期望最大化算法不将对象明确地分到某个簇，而是根据表示隶属可能性的权来分配对象。也就是说，在簇之间没有严格的边界。新的均值基于加权度量值计算。

在实际应用中，相当多的问题属于数据残缺问题，不能直接观察到的变量称为隐含变量，任

何含有隐含变量的模型都可以归为数据残缺问题。EM 算法是解决数据残缺问题的一个十分有效的算法。

令 $\boldsymbol{D}=\{\boldsymbol{x}(1),\boldsymbol{x}(2),\cdots,\boldsymbol{x}(n)\}$为 n 个观察到的数据向量组成的集合，$H=\{\boldsymbol{z}(1),\boldsymbol{z}(2),\cdots,\boldsymbol{z}(n)\}$表示隐含变量 Z 的 n 个值，分别与观察到的数据点一一对应，即 $\boldsymbol{z}(i)$与数据点 $\boldsymbol{x}(i)$相联系，$\boldsymbol{z}(i)$表示数据 $\boldsymbol{x}(i)$的不可见聚类标签。

可以把观察到的数据的对数似然写为

$$l(\theta)=\log p(D\mid\theta)=\log\sum_{H}p(D,H\mid\theta)$$

其中右侧的求和项表明，观察到的似然可以表示为观察到的数据和隐藏数据的似然对隐藏值的求和；$p(D,H|\theta)$为未知参数 θ 为参量的概率模型。

设 $Q(H)$为残缺数据 H 的任意概率分布，可以用以下方式表示似然：

$$\begin{aligned}l(\theta)&=\log\sum_{H}p(D,H\mid\theta)\\&=\log\sum_{H}Q(H)\frac{p(D,H\mid\theta)}{Q(H)}\geqslant\sum_{H}Q(H)\frac{p(D,H\mid\theta)}{Q(H)}\\&=\sum_{H}Q(H)\log p(D,H\mid\theta)+\sum_{H}Q(H)\log\frac{1}{Q(H)}\\&=F(Q,\theta)\end{aligned}$$

函数 $F(Q,\theta)$是要最大化的似然函数 $l(\theta)$的下限，算法重复以下两个步骤直至收敛：

（1）E 步骤：固定参数 θ，使 F 相对于分布 Q 最大化

$$Q^{k+1}=\arg\max_{Q}F(Q^{k},\theta^{k})$$

（2）M 步骤：固定分布 $Q(H)$，使 F 相对于参数 θ 最大化

$$\theta^{k+1}=\arg\max_{Q}F(Q^{k+1},\theta^{k})$$

可以证明，在 E 步骤中当 $Q^{k+1}=p(H\mid D,\theta^{k})$时似然达到最大值。对于这个 Q 值，不等式变成了等式：$l(\theta^{k})=F(Q,\theta^{k})$

在 M 步骤中，因为 F 中的第二项不依赖于 θ，最大化问题就简化为最大化 F 中的第一项，从而得到

$$\theta^{k+1}=\arg\max_{Q}\sum_{H}p(H\mid D,\theta^{k})\log p(H\mid D,\theta^{k})$$

在 E 步骤中，以参数向量 θ^{k}的特定设置为条件，估计隐藏变量的分布，在 M 步骤中，保持 Q 不变，选取新的参数 θ^{k+1}，使观察到的数据的期望对数似然最大化。通过 E 步骤和 M 步骤的迭代，求出收敛的参数解。

22.7　层次方法

层次聚类法又称树聚类算法，包括“自底向上”的凝聚法和“自顶向下”的分裂法。凝聚法

何含有隐含变量的模型都可以归为数据残缺问题。EM 算法是解决数据残缺问题的一个十分有效的算法。

令 $\boldsymbol{D}=\{\boldsymbol{x}(1),\boldsymbol{x}(2),\cdots,\boldsymbol{x}(n)\}$为 n 个观察到的数据向量组成的集合，$H=\{\boldsymbol{z}(1),\boldsymbol{z}(2),\cdots,\boldsymbol{z}(n)\}$表示隐含变量 Z 的 n 个值，分别与观察到的数据点一一对应，即 $\boldsymbol{z}(i)$与数据点 $\boldsymbol{x}(i)$相联系，$\boldsymbol{z}(i)$表示数据 $\boldsymbol{x}(i)$的不可见聚类标签。

可以把观察到的数据的对数似然写为

$$l(\theta)=\log p(D\mid\theta)=\log\sum_H p(D,H\mid\theta)$$

其中右侧的求和项表明，观察到的似然可以表示为观察到的数据和隐藏数据的似然对隐藏值的求和；$p(D,H|\theta)$为未知参数 θ 为参量的概率模型。

设 $Q(H)$为残缺数据 H 的任意概率分布，可以用以下方式表示似然：

$$\begin{aligned}
l(\theta)&=\log\sum_H p(D,H\mid\theta)\\
&=\log\sum_H Q(H)\frac{p(D,H\mid\theta)}{Q(H)}\geqslant\sum_H Q(H)\frac{p(D,H\mid\theta)}{Q(H)}\\
&=\sum_H Q(H)\log p(D,H\mid\theta)+\sum_H Q(H)\log\frac{1}{Q(H)}\\
&=F(Q,\theta)
\end{aligned}$$

函数 $F(Q，\theta)$是要最大化的似然函数 $l(\theta)$的下限，算法重复以下两个步骤直至收敛：

（1）E 步骤：固定参数 θ，使 F 相对于分布 Q 最大化

$$Q^{k+1}=\arg\max_Q F(Q^k,\theta^k)$$

（2）M 步骤：固定分布 $Q(H)$，使 F 相对于参数 θ 最大化

$$\theta^{k+1}=\arg\max_Q F(Q^{k+1},\theta^k)$$

可以证明，在 E 步骤中当 $Q^{k+1}=p(H\mid D,\theta^k)$时似然达到最大值。对于这个 Q 值，不等式变成了等式：$l(\theta^k)=F(Q,\theta^k)$

在 M 步骤中，因为 F 中的第二项不依赖于 θ，最大化问题就简化为最大化 F 中的第一项，从而得到

$$\theta^{k+1}=\arg\max_Q\sum_H p(H\mid D,\theta^k)\log p(H\mid D,\theta^k)$$

在 E 步骤中，以参数向量 θ^k 的特定设置为条件，估计隐藏变量的分布，在 M 步骤中，保持 Q 不变，选取新的参数 θ^{k+1}，使观察到的数据的期望对数似然最大化。通过 E 步骤和 M 步骤的迭代，求出收敛的参数解。

22.7　层次方法

层次聚类法又称树聚类算法，包括“自底向上”的凝聚法和“自顶向下”的分裂法。凝聚法

先将所有对象各自作为簇，将最“靠近”的簇首先进行聚类，再将这个类和其他类中最“接近”的簇合并，该过程递归进行直至所有对象都聚集成一个簇或满足一个终止条件为止。分裂法正好相反，先将所有对象看成一个簇，然后割成两个，使一个簇中的对象尽可能“远离”另一个簇中对象，再递归分割，直至每个对象都自成一个簇或满足某个终止条件为止。

凝聚或分裂的过程可用树形图直观表示，该图显示簇—子簇联系和簇合并（凝聚）或分裂的次序。在层次聚类方法中，距离定义非常重要，簇间距离描述两类簇关系，比较常用的定义有如下几种。

（1）最短距离（单连接方法）

$$d_{\min}(C_i,C_j)=\min_{p\in Ci,p'\in C_j}\|p-p'\|$$

（2）最长距离（完全链接方法）

$$d_{\max}(C_i,C_j)=\min_{p\in Ci,p'\in C_j}\|p-p'\|$$

（3）中间距离（平均链接方法）

$$d_{avg}(C_i,C_j)=\frac{1}{n_in_j}\sum_{p\in Ci}\sum_{p'\in C_j}\|p-p'\|$$

（4）均值距离（质心方法）

$$d_{mean}(C_i,C_j)=\|m_i-m_j\|$$

对象间距离函数有欧氏距离、闵可夫距离、马氏距离等，同样地，簇间距离或相似度也有多种选择，不同的距离函数可以得到不同的层次聚类方法。图 22.2 给出了凝聚的和分裂的层次聚类方法的处理过程。

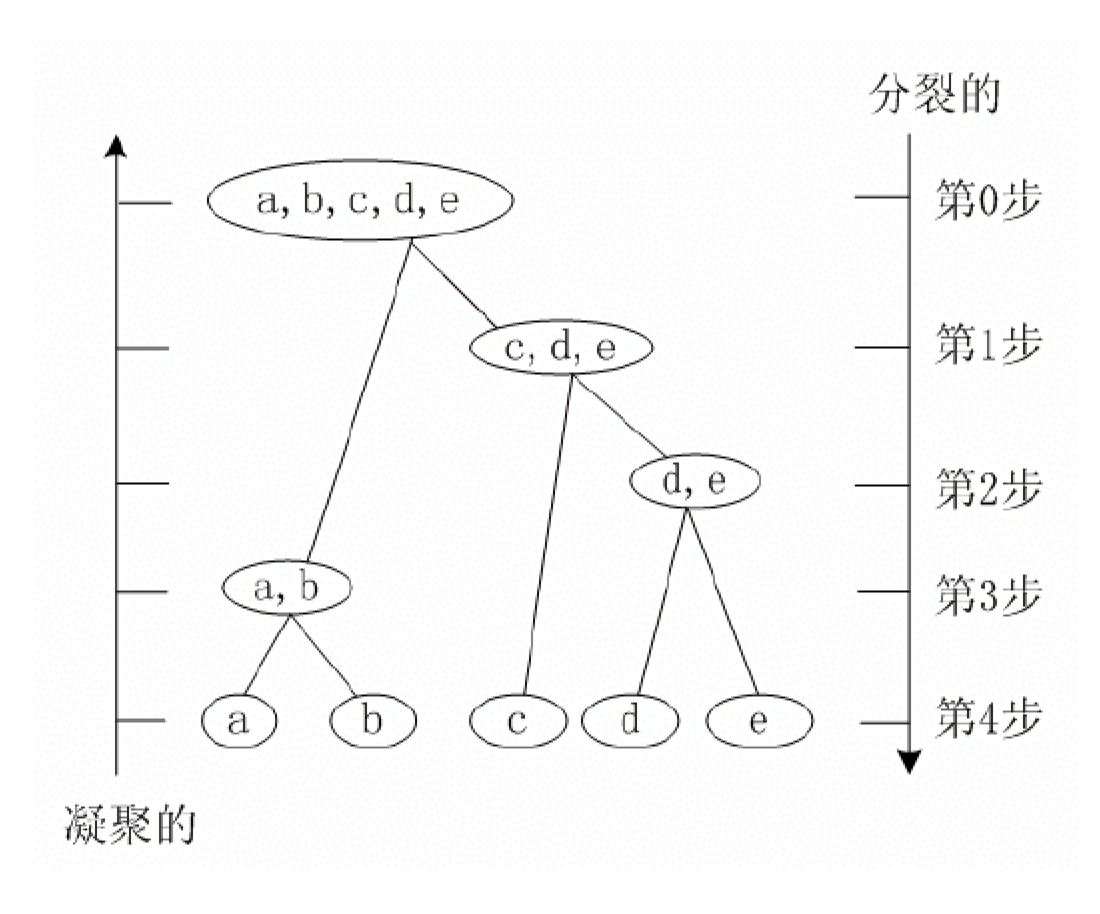


图 22.2　凝聚的和分裂的层次聚类方法

层次聚类方法的优点在于可以在不同粒度水平上对数据进行探测，而且容易实现相似度量或距离度量，但是，单纯的层次聚类算法终止条件含糊（一般需人为设定），而且执行合并或分裂簇的操作后不可修复，这很可能导致聚类结果质量很低。由于需要检查和估算大量的对象或簇才能决定簇的合并或分裂，所以这种方法的可扩展性较差。因此，通常考虑把层次聚类方法与其他方法如迭代重定位方法相结合来解决实际聚类问题。

层次聚类和其他聚类方法的有效集成可以形成多阶段聚类，能够改善聚类。

22.7.1 利用层次方法的平衡迭代归约及聚类

对于海量数据的聚类问题，可以用 BIRCH 算法来处理。该方法是一种非常有效的聚类技术，用于欧氏空间数据，即平抑值有意义的数据，算法单遍扫描数据集就可生成较好的聚类，一遍或多遍的扫描可以改进聚类质量。

聚类特征和 CF-树是 BIRCH 算法的关键，聚类特征的定义如下。

在一个簇中给定 N 个 d-维数据点 $\{X_i\}(i=1,2,\cdots,N)$，聚类特征定义为一个三元组 $\mathrm{CF}=(N,\overrightarrow{\mathrm{LS}},\mathrm{SS})$，其中 N 是聚类中数据点的数量，$\overrightarrow{\mathrm{LS}}$ 是 N 个数据点的线性和，即 $\sum_{i=1}^{N}X_i$，SS 是 N 个数据点的平方和，即 $\sum_{i=1}^{N}X_i$。

聚类特征具有加和性，假设有两个不交的簇的聚类特征分别为 $\mathrm{CF}_1=(N_1,\overrightarrow{\mathrm{LS}_1},\mathrm{SS}_1)$ 和 $\mathrm{CF}_2=(N_2,\overrightarrow{\mathrm{LS}_2},\mathrm{SS}_2)$，由这两个合并形成的新的簇的聚类特征为

$$\mathrm{CF}=\mathrm{CF}_1+\mathrm{CF}_2=(N_1+N_2,\overrightarrow{\mathrm{LS}_1}+\overrightarrow{\mathrm{LS}_2},\mathrm{SS}_1+\mathrm{SS}_2)$$

从而可知，簇的 CF 是可存储的，而且在合并簇或加入新数据点时，CF 是可进行增量计算的。

簇是数据点的集合，BIRCH 算法中只存储聚类特征汇总。与存储簇内所有点的信息相比，存储聚类特征汇总不仅效率高，而且可以保证准确度。

CF 树存储了层次聚类的聚类特征，它是一棵带有两个参数的高度平衡的树，这两个参数为分支因子 B 和阈值 T。其中，非叶子节点至多有 B 个形如$[\mathrm{CF}_i,\mathrm{child}_i](i=1,2\cdots,B)$的项，$\mathrm{child}_i$ 是指向第 i 个子代节点的指针，CF_i 是该子代节点表示的子簇的聚类特征，非叶子节点表示由所有子代节点表示的子簇组合形成的簇。叶子节点至多包含 L 个形如$[\mathrm{CF}_i](i=1,2\cdots,L)$项。另外。叶子节点还有两个指针“*prev*”和“*next*”，用于把所有叶子连成链，达到高效扫描的目的。叶子节点表示由相应项描述的子簇形成的簇。叶子节点的项应该满足阈值 T，T 表示叶子节点中子聚类的最大直径（或半径）。

由于叶子节点中的项是子簇而不是单个数据点，因此，CF 树是对聚类数据的简洁表示，显然参数 B 和 T 决定 CF 树的规模。

当插入新数据对象是地，CF 树可以动态构造。CF 树的重建类似于树构建中的节点插入和节点分裂。

采用 CF 及 CF 树结构有利于增量聚类和动态聚类。BIRCH 采用多阶段聚类技术，对数据集合进行单遍扫描后生成初步簇，再经过一遍或多遍扫描改进聚类。算法的复杂度为 $O(n)$，其缺点在于 CF 树对节点中包含项的数目有限制，这可能导致节点并未对应实际数据集的一个自然簇。

BIRCH 算法主要分四个阶段：第一阶段对整个数据集进行扫描，根据给定的初始距离阈值建立一棵初始聚类特征树；第二阶段通过提升阈值 T 重建 CF 树，得到一棵压缩的 CF 树。第三、第四阶段利用全局聚类算法对已有的 CF 树进行聚类得到更好的聚类结果。

BIRCH 算法利用聚类特征树概括了聚类的有用信息，并且由于聚类特征树占用空间比原始数据集合小得多，可以存入内存中，因此在给定有限内存的情况下，BIRCH 能利用可用的资源产生较好的聚类结果。算法的复杂度为 $O(N)$，具有与对象数目呈线性关系的可扩展性和较好的聚类质量。但是由于大小限制，CF 树的每个节点只能包含有限数目的项目，一个 CF 树节点并

不总是对应于用户所考虑的一个自然簇。此外，由于采用直径或半径 T 来控制聚类边界，BIRCH 算法不适合发现非球形的簇。

22.7.2 利用代表点聚类

CURE（利用代表点聚类）算法是介于基于质心方法和基于代表对象点方法之间的策略。在 CURE 算法中，不是利用质心或单个代表对象点来代表一个簇，而是首先在簇中选取固定数目的、离散分布的点，用这些点反映簇的形状和范围。然后把离散的点按收缩因子向簇的质心收缩。收缩后的离散点作为簇的代表点。两个簇的距离定义为代表点对（分别来自两个簇）距离的最小值，在 CURE 算法的每一步合并距离最近的两个簇。

CURE 算法克服了利用单个代表点或基于质心方法的缺点，可以发现非球形及大小差异较大的簇。簇的收缩（离散点的收缩）降低了算法对孤立点的敏感性。

调节收缩因子 α（$\alpha \in [0,1]$），可以让 CURE 发现不同形式的簇。当 $\alpha = 1$ 时，CURE 还原为基于质心的方法。当 $\alpha = 0$ 时，CURE 还原为 MST（最小生成树）方法。

在大数据集的聚类问题上，CURE 采取随机采样的方法。虽然随机采样是在精确与效率间的折中，实验证明，对于多数数据集，中等规模的采样就能较好地保证聚类质量。为加速聚类收敛速度，CURE 算法首先对样本数据进行划分并在每个划分块内局部聚类，去除孤立点后，再对每个划分块中局部的簇进行聚类生成最后的簇。

对于容量为 n 的样本，CURE 算法的最差时间复杂度为 $O(n^2\log n)$。当数据点维数较低时（如 2 维），时间复杂度可减少为 $O(n^2)$，该算法仅对数据库扫描一遍，其空间复杂度为 $O(n)$。

22.8 基于密度的方法

基于密度聚类的关键思想是：对于簇中每个对象，在给定半径 ε 的邻域中至少要包含最小数目的对象($MinPts$)，即邻域的基数必须超过一个阈值。基于密度的方法主要有两类，即基于连通性的算法和基于密度函数的算法。基于连通性的算法包括 DBSCAN、GDBSCAN、OPTICS、DBCLASD 等；基于密度函数的算法有 DBNCLUE 等算法。

大型空间数据库中可能含有球形、线形、延展形等多种形状的簇，因此，要求聚类算法应具有能够发现任意形状簇的能力。当然还要求聚类算法在大型数据库上具有高效性。DBSCAN 算法就是满足上述要求的一种基于密度的聚类算法，它将足够高密度的区域划分为簇，能够在含有“噪声”的空间数据库中发现任意形状的簇。点邻域的形状取决于两点间的距离函数 $dist(p,q)$。例如采用二维空间的曼哈坦距离时，邻域的形状为形状。在实际应用中应该采用能反映问题特性的距离函数。

基于密度的簇和“噪声”的概念是基于下列各定义。

定义 I：点 p 的 ε-邻域可记为 $N_\varepsilon(p)$，其定义为

$$N_\varepsilon(p) = \{q \in D \mid \mathrm{dist}(p,q) \leqslant \varepsilon\}$$

定义 II：如果 p、q 满足下列条件：（1）$p \leqslant N_\varepsilon(p)$，（2）$|N_\varepsilon(q)| \geqslant \mathrm{MinPts}$，则称点 p 是从点 q 关于 ε 和 MinPts 直接密度可达的。

显然，直接密度可达关系在核心点对间是对称的。在核心点和边界点间直接密度可达关系不是对称的，如图 22.3 所示。

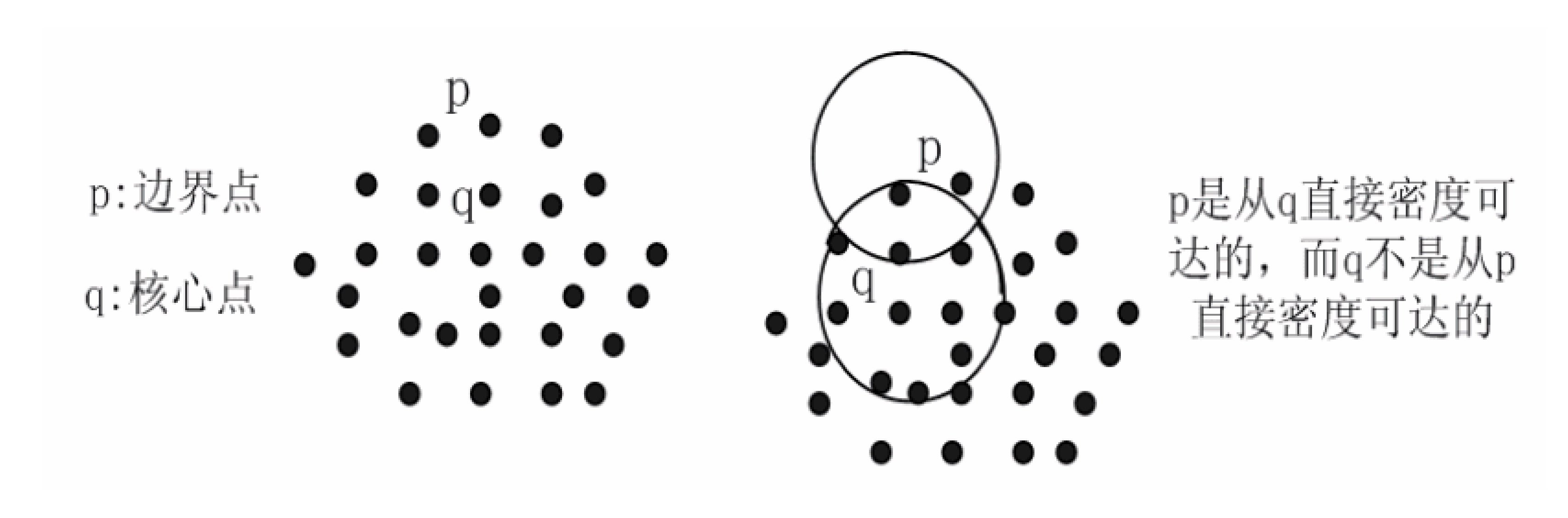


图 22.3　核心点、边界点、直接密度可达

定义 III：如果存在一个点的序列 $p_1,p_2\cdots,p_n$，$p_1=q,p_n=p,p_{i+1}$ 是从 p_i 直接密度可达的，则称点 p 是从点 q 关于 ε 和 MinPts 密度可达的。

密度可达是直接密度可达的扩展，密度可达关系满足传递性，但不满足对称性。

定义 IV：如果存在一个点 o，p 和 q 都是从点 o 关于 ε 和 MinPts 密度可达的，则称点 p 是从点 q 关于 ε 和 MinPts 密度相连的。

密度相连是一个对称关系，密度可达的点之间的密度相连关系还满足自反性。

在上述 4 个定义的基础上，就可以定义基于密度的簇和"噪声"的概念。

簇的定义：令 D 表示数据点的集合，若 D 的非空子集 C 满足下列条件。

（1）对任意 p 和 q，若 $p\in C$ 且 q 是从 p 关于 ε 和 MinPts 密度可达的，则有 $q\in C$。（最大性）

（2）$\forall p,q\in C$：p 与 q 是关于 ε 和 MinPts 密度相连的。（连通性）

则称 C 是基于密度簇。它是基于密度可达的最大的密度相连的点的集合。

噪声的定义：令 $C_1,C_2,\cdots,C_K$ 是数据库中分别关于参数 ε_i 和 MinPts_i 构成的簇，则"噪声"被定义为数据库中不属于任何簇的数据点的集合，即集合 $\{p\in D\,|\,\forall_i:q\notin C_i\}$ 就为"噪声"。

给定参数 ε 和 MinPts，可以分两步发现簇。第一步，从数据库中任意选取一个满足核心点的点作为种子；第二步，检索从种子点密度可达的所有点，获得包括种子点在内的簇。

DBSCAN 算法可以发现空间数据中的簇和"噪声"。但必须为每个簇指定恰当的参数 ε 和 MinPts，及至少每个簇中的一个点。但要事先获得数据库中所有簇的相关信息并不是一件容易的事。

为发现簇，DBSCAN 算法从任意点 p 开始，检索所有从点 p 关于 ε 和 MinPts 密度可达的点。如果 p 是核心点，就生成一个关于 ε 和 MinPts 的簇；如果 p 是边界点，且没有从 p 密度可达的点，DBSCAN 算法就访问数据库中下一个点。由于 ε 和 MinPts 是全局参数值，如果两个不同密度的簇彼此接近，DBSCAN 可能会合并这两个簇。当没有新的点添加到任何簇时，过程结束。

虽然 DBSCAN 算法可以对数据对象进行聚类，但需要由用户确定输入参数 ε 和 MinPts，而且算法对参数值非常敏感，参数值的微小变化往往会导致差异很大的聚类结果，所以在精确地确定这两个参数，但在现实的高维数据集合中，很难做到这一点。

OPTICS 算法为自动和交互的聚类分析提供了一个可扩展的簇次序。簇次序刻画了表达数据集的基于密度的聚类结构，它包含的信息等价于一个参数设定范围宽广的基于密度的聚类。簇次序可作为自动和交互聚类的基础。

DENCLUE 算法是基于密度函数的聚类方法，它的基本思想是把每一个数据点对聚类的影响利用数学函数形式化地建模，这些数学函数称为影响函数。影响函数可以看作描述数据点在其邻

域内的影响程度，抛物线函数、方波函数、高斯函数等都可以作为影响函数。数据空间的整体密度可以通过所有点影响函数的加和计算得出，然后通过确定密度吸引点的方法精确地确定簇。密度吸引点是全局密度函数的局部最大值。如果全局密度函数是连续的且在任意点可导，就可以用全局密度函数的梯度指导爬山算法有效地确定密度吸引点。

22.9 基于网格的方法

基于网格的方法首先将空间量化为有限数目的单元，然后在这个量化空间上进行所有的聚类操作。这类方法的处理时间不受数据对象数目的影响，仅依赖于量化空间中每一维上的单元数目，因此处理速度较快。

STING 算法即是一种基于网络的方法，它利用层次结构将空间区域划分为矩形单元，在每个单元中存储对象的统计参数（如均值、方差、最小值、最大值、分布的类型等），用以描述有关数据特征。STING 通过对数据集进行一次扫描，计算单元中的统计参数。因此，若 n 表示对象的个数，则生成簇的时间复杂度为 $O(n)$。

在生成层次结构后，一个查询的响应时间是 $O(k)$。其中 k 是最低分辨率下网络单元的数目，通常 k 远小于 n。STING 采用多分辨率的方式进行聚类，聚类质量取决于网络结构中底层的粒度。

WaveCluster 算法利用小波变换聚类，该算法既是基于网络的，也是基于密度的，其主要思想是，首先量化特征空间，形成一个多维网络结构，然后通过小波变换来变换原始特征空间，最后在变换后的特征空间中发现密集区域。它可以有不同分辨率下产生基于用户需求的簇。

WaveCluster 算法中的每个网络单元汇总一组映射到该单元的对象的信息。这种汇总信息可以用于基于内存的多分辨率小波变换，以及随后的聚类分析。

WaveCluster 算法的第一步是量化特征空间。把 d 维特征空间的第 i 维分割成个 m_i 区间。如果假定各个维上的区间数等于 m，那么，在特征空间中将有 m^d 个单元，然后，根据特征值将对象分配到这些单元中。令 $\boldsymbol{F}_k=(f_1,f_2,\cdots,f_d)$ 为对象 O_k 在原始特征空间中的特征向量。$M_j=(v_1,v_2,\cdots,v_d)$ 表示原始特征空间中的一个单元，其中 $v_i(1\leqslant v_i\leqslant m_i$，$1\leqslant i\leqslant d)$ 是该单元在特征空间的 X_i 轴上的位置。令 s_i 为 X_i 轴上每个单元的大小，如果具有特征向量 $\boldsymbol{F}_k=(f_1,f_2,\cdots,f_d)$ 的对象 O_k 满足对 $\forall i$，$1\leqslant i\leqslant d$ 有：$(v_i-1)s_i\leqslant f_i\leqslant v_i s_i$，则将该对象分配到单元 $M_j=(v_1,v_2,\cdots,v_d)$。

单元的数目是影响聚类质量的一个重要因素，由于小波变换具有多分辨率特性，所以要在不同的变换尺度上考虑不同的单元大小。

WaveCluster 算法的第二步是对特征空间进行小波变换。离散小波变换应用于量化的特征空间。在单元 M_j 上应用小波变换产生新的特征空间和新的单元 T_k，给定单元 T_k 的集合。WaveCluster 在变换后的特征空间发现相连的部分，每一个相连的部分是单元 T_k 的集合，将它们看成是簇。对应小波变换的分辨率 r，存在簇的集合 C_r，通常较粗的分辨率对应的簇的数目较少。

任意一个簇 c，$c\in C_r$，c 含有的簇数目为 c_n。在 WaveCluster 算法的第 4 步标记特征空间中的单元。WaveCluster 用单元中簇的数目来标记特征空间中含有簇的单元，即

$$\forall c\forall T_k, T_k\in c\Rightarrow l_{T_k}=c_n, c\in C_r$$

式中：l_{T_k} 是单元 T_k 的标记。簇是在变换后的特征空间中发现的，而且是基于小波系数的，因此，不能直接用于定义原始特征空间中的簇。WaveCluster 生成一个查询表 LT，将变换后的特征空间

域内的影响程度，抛物线函数、方波函数、高斯函数等都可以作为影响函数。数据空间的整体密度可以通过所有点影响函数的加和计算得出，然后通过确定密度吸引点的方法精确地确定簇。密度吸引点是全局密度函数的局部最大值。如果全局密度函数是连续的且在任意点可导，就可以用全局密度函数的梯度指导爬山算法有效地确定密度吸引点。

22.9 基于网格的方法

基于网格的方法首先将空间量化为有限数目的单元，然后在这个量化空间上进行所有的聚类操作。这类方法的处理时间不受数据对象数目的影响，仅依赖于量化空间中每一维上的单元数目，因此处理速度较快。

STING 算法即是一种基于网络的方法，它利用层次结构将空间区域划分为矩形单元，在每个单元中存储对象的统计参数（如均值、方差、最小值、最大值、分布的类型等），用以描述有关数据特征。STING 通过对数据集进行一次扫描，计算单元中的统计参数。因此，若 n 表示对象的个数，则生成簇的时间复杂度为 $O(n)$。

在生成层次结构后，一个查询的响应时间是 $O(k)$。其中 k 是最低分辨率下网络单元的数目，通常 k 远小于 n。STING 采用多分辨率的方式进行聚类，聚类质量取决于网络结构中底层的粒度。

WaveCluster 算法利用小波变换聚类，该算法既是基于网络的，也是基于密度的，其主要思想是，首先量化特征空间，形成一个多维网络结构，然后通过小波变换来变换原始特征空间，最后在变换后的特征空间中发现密集区域。它可以有不同分辨率下产生基于用户需求的簇。

WaveCluster 算法中的每个网络单元汇总一组映射到该单元的对象的信息。这种汇总信息可以用于基于内存的多分辨率小波变换，以及随后的聚类分析。

WaveCluster 算法的第一步是量化特征空间。把 d 维特征空间的第 i 维分割成个 m_i 区间。如果假定各个维上的区间数等于 m，那么，在特征空间中将有 m^d 个单元，然后，根据特征值将对象分配到这些单元中。令 $\boldsymbol{F}_k=(f_1,f_2,\cdots,f_d)$为对象 O_k 在原始特征空间中的特征向量。$M_j=(v_1,v_2,\cdots,v_d)$表示原始特征空间中的一个单元，其中 $v_i(1\leqslant v_i\leqslant m_i$，$1\leqslant i\leqslant d)$是该单元在特征空间的 X_i 轴上的位置。令 s_i 为 X_i 轴上每个单元的大小，如果具有特征向量 $\boldsymbol{F}_k=(f_1,f_2,\cdots,f_d)$的对象 O_k 满足对 $\forall i$，$1\leqslant i\leqslant d$ 有：$(v_i-1)s_i\leqslant f_i\leqslant v_i s_i$，则将该对象分配到单元 $M_j=(v_1,v_2,\cdots,v_d)$。

单元的数目是影响聚类质量的一个重要因素，由于小波变换具有多分辨率特性，所以要在不同的变换尺度上考虑不同的单元大小。

WaveCluster 算法的第二步是对特征空间进行小波变换。离散小波变换应用于量化的特征空间。在单元 M_j 上应用小波变换产生新的特征空间和新的单元 T_k，给定单元 T_k 的集合。WaveCluster 在变换后的特征空间发现相连的部分，每一个相连的部分是单元 T_k 的集合，将它们看成是簇。对应小波变换的分辨率 r，存在簇的集合 C_r，通常较粗的分辨率对应的簇的数目较少。

任意一个簇 c，$c\in C_r$，c 含有的簇数目为 c_n。在 WaveCluster 算法的第 4 步标记特征空间中的单元。WaveCluster 用单元中簇的数目来标记特征空间中含有簇的单元，即

$$\forall c \forall T_k, T_k \in c \Rightarrow l_{T_k} = c_n, c \in C_r$$

式中：l_{T_k} 是单元 T_k 的标记。簇是在变换后的特征空间中发现的，而且是基于小波系数的，因此，不能直接用于定义原始特征空间中的簇。WaveCluster 生成一个查询表 LT，将变换后的特征空间

中的单元映射到原始特征空间中的单元。查询表中的每个元素表示在变换后的特征空间中一个单元与原始特征空间中的相应单元的关系。因此，能够很容易地确定原始特征空间中的每个单元的标记。最后，WaveCluster 将特征空间中每个单元的标记，分配给所有特征向量在该单元中的对象，从而确定了簇。形如

$$\forall c \forall M_j, \forall o_i \in M_j, l_{o_i} = c_n, c \in C_r, 1 \leqslant i \leqslant N$$

其中：l_{o_i} 是对象 o_i 的簇标记。

WaveCluster 算法能够较好地处理孤立点，对输入数据的顺序不敏感，对大型数据库有效，它能够较好地发现带有不同比例的复杂结构（如凹形的或窝形的）的簇，且不需要为簇假定任何特征的形状，不要求簇的数目等先验知识。

22.10　基于模型的聚类方法

基于模型的聚类方法建立在数据符合潜在的概率分布这一假设基础之上。该类方法试图优化给定数据与某些数学模型之间的拟合，主要有统计学方法和神经网络方法等。

COBWEB 是一种简单增量概念聚类算法，它以一个分类树的形式创建层次聚类。分类树与判定树不同，分类树中每一个节点对应一个概念，包含该概念的一个概率描述，概括该节点的对象信息。判定树标记分支而不是节点，并且采用逻辑描述符，而不是概率描述符。COBWEB 采用启发式估算量度—分类效用来指导分类树的构建，如果要将对象加入对象树，就要加入到能产生最高分类效用的位置，即根据产生最高分类效用的划分，把对象置于一个存在的类中，或者为它创建一个新类。COBWEB 可以自动修正划分中类的数目，不需要用户提供相应参数。但它的局限性在于假设每个属性上的概率分布相互独立，而实际上属性常常是相关的。另外，聚类的概率分布表示使得更新和存储聚类代价较高。算法的计算复杂度不仅依赖于属性数目，而且依赖于属性值的数目。分类树在偏斜的数据上难以达到高度平衡，这可能导致时间和空间复杂度的剧烈变化。

CLASSIT 对 COBWEB 进行扩展，用来处理连续性数据的增量聚类。该算法在每个节点中存储属性的连续正态分布，采用修正的分类效用度量，该度量是连续属性上的积分，而不是在离散属性上求和。但 CLASSIT 存在与 COBWEB 类似的问题，也不适用于对大型数据库中的数据进行聚类。

AutoClass 是在产业界较为流行的聚类方法，它采用贝叶斯统计分析来估算结果簇的数目。该系统通过搜索模型空间所有的分类可能性，自动确定分类类别的个数和模型描述的复杂性。它允许在一定的类别内属性之间具有一定的相关性，各个类之间具有一定的继承性，即在类层次结构中，某些类共享一定的模型参数。

神经网络方法将每个簇描述为一个样本。样本作为聚类的原型，不一定对应特定的数据实例和对象。神经网络聚类方法包括 Rumelhart 等人提出的竞争学习神经网络和 Kohonen 提出的自组织特征映射(SOM)神经网络。神经网络聚类方法处理时间较长，并用有较高的数据复杂性。需要研究提高网络学习速度的学习算法，并增强网络的可理解性，以便使人工神经网络适用于大型数据库。

22.11 基于目标函数的方法

前面各节提到的算法一般都为适用于动态数据库的聚类技术。实际中受到人们普遍欢迎的是基于目标函数的聚类方法，该方法将聚类归结成一个带约束的非线性规划问题，通过优化技术获得数据集的划分和聚类。这类方法设计简单、解决问题的范围广，还可以转化为优化问题而借助经典数学的非线性规划理论求解，并易于在计算机上实现。因此随着计算机的应用和发展，基于目标函数的聚类算法成为新的研究热点。

设有两个样本 $\boldsymbol{X}_i$、$\boldsymbol{X}_j$ 的特征向量分别为

$$X_i=\begin{bmatrix}x_{i1}\\x_{i2}\\\vdots\\x_{in}\end{bmatrix}=(x_{i1},x_{i2},\cdots,x_{in})^T,\quad X_j=\begin{bmatrix}x_{j1}\\x_{j2}\\\vdots\\x_{jn}\end{bmatrix}=(x_{j1},x_{j2},\cdots,x_{jn})^T$$

这两个样本可以在同一类中，也可能在不同的类中，因此可以计算同一个类内样本与样本之间的距离，也可以计算属于不同类的样本与样本之间的距离。

计算样本与样本间的距离有几种方法，分别是欧氏距离法、夹角余弦距离法、二值夹角余弦法和具有二值特征的 Tanimoto 测度等。

欧氏距离：$D_{ij}^2=\|X_i-X_j\|^2=(X_i-X_j)^T(X_i-X_j)=\sum_{k=1}^{n}(x_{ik}-x_{jk})^2$

马氏距离：
$$D_{ij}^2=(X_i-X_j)^T S^{-1}(X_i-X_j)$$
$$S=\frac{1}{N-1}\sum_{i=1}^{n}(X_i-\overline{X})(X_i-\overline{X})^T\qquad \mathrm{X}=\frac{1}{\mathrm{N}}\sum_{i=1}^{N}\mathrm{X}_i$$

夹角余弦距离：$S(X_i,X_j)=\cos\theta=\dfrac{X_iX_j}{\|X_i\|\bullet\|X_j\|}$

特征是二值时的夹角余弦：$S(X_i,X_j)=\cos\theta=\dfrac{X_i^TX_j}{\sqrt{(X_i^TX_i)(X_j^TX_j)}}$

具有二值特征的 Tanimoto 测度：$S(X_i,X_j)=\dfrac{X_i^TX_j}{X_i^TX+X_j^TX_j-X_i^TX_j}$

22.11.1 样本与类之间的距离

ω 是代表某样本的集合，ω 中有 N 个样本，$\boldsymbol{X}$ 是某一个待测样本。

样本与类之间的距离的计算方法有两种。

（1）计算该样本到 ω 类内各个样本之间的距离，将这些距离求和，然后取平均值作为样本与类之间的距离

$$\overline{D^2(X,\omega)}=\frac{1}{N}\sum_{i=1}^{N}D^2(X,X_i^{(\omega)})=\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{n}|x_k-x_{ik}^{(\omega)}|^2$$

（2）计算 ω 类的中心点 $\boldsymbol{M}^{(\omega)}$，以 ω 中的所有样本特征的平均值作为类中心，然后计算待

测样本 $\boldsymbol{X}$ 到 ω 的中心点 $\boldsymbol{M}^{(\omega)}$ 的距离。

$$D^2(X,\omega)=D^2(X,M^{(\omega)})=\sum_{k=1}^{n}|x_k-m_k^{(\omega)}|^2$$

22.11.2 类内距离

类内距离是指同一个类内任意样本之间的距离之和的平均值。从类集内一固定点 $\boldsymbol{X}_i$ 到所有其他的 $N-1$ 个点 $\boldsymbol{X}_j$ 之间的距离平方和为

$$\overline{D^2(X_i,\{X_j\})}=\frac{1}{N-1}\sum_{\substack{j=1\\j\neq i}}^{N}\sum_{k=1}^{n}(x_{ik}-x_{jk})^2$$

同样道理，取 ω 内所有 N 个点的平均距离以表示其类内距离

$$\overline{D^2(\{X_i\},\{X_j\})}=\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}\sum_{k=1}^{n}(x_{ik}-x_{jk})^2$$

22.11.3 类与类之间的距离

设有两个类 ω_i、ω_j，计算类与类之间的距离有以下几种方式。

（1）最短距离法：规定两个类间相距最近的两个点之间的距离为两类的距离

$$D_{i,j}=\min(d_{ij})$$
$$d_{ij}=\|X_i-X_j\|,X_i\in\omega_i,X_j\in\omega_j$$

（2）最长距离法：规定两个类间相距最远的两个点之间的距离为两类的距离

$$D_{i,j}=\max(d_{ij})$$
$$d_{ij}=\|X_i-X_j\|,X_i\in\omega_i,X_j\in\omega_j$$

（3）重心法：求各类中所有样本的平均值作为类的重心，用两类的重心间的距离作为两个的距离

$$D_{i,j}=\|\overline{X^{(\omega_i)}}-\overline{X^{(\omega_j)}}\|$$
$$\overline{X^{(\omega_i)}}=\frac{1}{N_i}\sum_{X\in\omega_i}X,\overline{X^{(\omega_j)}}=\frac{1}{N_j}\sum_{X\in\omega_j}X$$

（4）平均距离法：计算两类之间所有样本的距离，求和，取距离的平均值作为两类间的距离

$$D_{i,j}=\frac{1}{N_iN_j}\sum_{\substack{X_i\in\omega_i\\X_j\in\omega_j}}\|X_i-X_j\|$$

根据以上各距离的计算方法，就可以构造聚类时的目标函数，一般要求类间距离要大，而每类样本间的距离要小。

聚类时的目标函数设定后，就可以利用各种优化方法对其进行求解，从而可得到聚类结果。具体的优化方法可参见第 3 篇“目标优化技术”相关内容。

22.12 离群点检测

离群点（Outlier）是指数据集合中不符合数据一般特性或一般模型的数据对象，又称孤立点。离群点可能是由于度量或执行错误产生的，也有可能是由于固有数据的变异产生的。

很多数据挖掘算法尽量减少离群点对挖掘结果的影响，或者在挖掘过程中排除离群点。但是，有时离群点（噪声）可能是非常重要的，识别离群点的模式比正常数据的模式更有价值，一味地排除离群点或降低孤立点的影响，将有可能导致丢失隐藏的重要信息。例如，在商业欺诈探测中，离群点可能预示着欺诈行为。在这种情况下，离群点的探测和分析是主要的挖掘任务，称为离群点挖掘。

离群点检测目前已成为数据挖掘的一个重要方面，正在得到越来越广泛的应用，在许多应用领域（如风险控制领域），特别是在“广义安全问题”中，离群点检测正逐步成为一种有用的工具，被用来发现稀有模式，或数据集中明显不同于其他数据的对象。通过对离群点的分析可以迅速、准确地甄别异常事件，如电信、保险、银行、电子的欺诈检测，灾害气象预报，商业营销中的特殊客户分析，医学诊断研究中发现新的疾病，医疗方案或药品所产生的异常反应，网络安全中的入侵检测，海关报关中的价格隐瞒，天文学中一些稀有的、新类型天体的发现，运动员的成绩分析、过程控制中的故障检测与诊断及文字编辑系统的设计等。

离群点可能由测量、输入错误或系统运行错误而造成，也可以是数据内存特性所决定的，或因客体的异常行为所导致的。由于离群点产生的机制是不确定的，离群点检测算法检测出的“离群点”是否真正对应实际的异常行为，不是由离群点检测算法而只能由领域专家来解释。算法只能从数据体现的规律角度为用户提供可疑的数据，以便引起用户特别的注意并最后确定是否为真正的异常。对于离群点的处理方式也取决于应用，并由领域专家决策。

对于给定的 n 个数据对象集合上的离群点挖掘，是指发现与其余数据相比有显著差异、异常或不一致的 k 个对象。首先要在给定的数据集合中定义数据的不一致性，然后找到有效的方法来挖掘离群点。

离群点的定义是非平凡的，如果采用一个回归模型，偏差分析可以给出对数据“极端性”的估计。但是，在时间序列数据中寻找离群点十分困难，它们可能隐藏在带趋势的、季节性的或者其他周期性变化中。当分析多维数据时，具有极端性的可能是维值的组合，而不是某个特别维值，对于非数值型的数据（如分类数据），离群点的定义建立在特殊的考虑基础之上。

由于人眼只善于识别至多三维的数值型数据，所以利用现有的数据可视化方法来分析很多分类属性的数据或高维数据中的离群点是低效率的。现在一般采用计算机技术。基于计算机的离群点探测方法可分为统计学方法、基于距离的方法、基于偏移的方法等。

统计学方法假定数据服从一定的概率分布或概率模型，然后根据模型采用不一致性检验来识别离群点。不一致性检验需要数据集参数（假定的数据分布）、分布参数（如均值和方差）及期望得到的离群点数目。基于统计学方法的离群点检测的主要缺点在于大多数检验是针对单个属性的，而许多数据挖掘问题要在高维数据空间中发现离群点。另外，统计学方法需要数据集合参数，例如数据分布，但同样在现实中数据分布也可能是未知的。因此在没有特定检验时，统计学方法

不能确定能发现所有的离群点。

为消除统计学方法带来的缺陷，引入基于距离的离群点检测的概念。若数据集 S 中至少有 p 个部分与对象 o 的距离大于 d，则对象 o 是一个在参数 p 和 d 下的基于距离的离群点，即在基于距离的离群点检测中，将离群点看作是那些没有足够数量邻居的对象。与基于统计的方法相比，基于距离的离群点检测拓宽了多个标准分布的不一致性检验的思想，避免了过多运算。常用的基于距离的离群点检测方法有基于索引的算法、嵌套—循环算法，基于单元的算法等。

基于偏离的离群点检测将离群点定义为与给定的描述偏离的对象。该类方法不采用统计检验或基于距离的度量来确定异常对象，而是通过检查一组对象的主要特征来确定离群点，序列异常技术和 OLAP 数据立方体技术是两种常见的基于偏离的离群点探测技术。

离群点检测中需要注意以下几个问题：

（1）全局观点和局部观点。

离群点与众不同，但具有相对性。一个对象可能相对于所有对象是离群的，但它相对于它的局部近邻不是离群的。

（2）点的离群程度。

某些技术方法以二值方法来报告对象是否为离群点，但这不能反映某些对象比其他对象更加偏离整体的基本事实。这时可以通过定义对象的偏离程度来给对象打分即离群因子或离群值得分，即在都为离群点的情况下，也还有分高和分低的区别。

（3）离群点的数量及检测的时效性。

数据集中离群点的数量通常是未知的，正常点的数量远远超过离群点的数量，离群点的数量在大规模数据集中所占的比例较低，一般小于 5%甚至 1%。

离群点在整个数据集中的比例很低，从数据是否偏离整体的角度看，这是一类极端不平衡的问题，离群点的检测可以看成一类极端不平衡的数据分类问题，但由于分布的极端不平衡，因此通常的分类方法、不平衡分类方法难以适用。

以下为常用的离群点检测方法。

22.12.1　基于统计的离群点检测方法

基于统计的方法是研究最早也是研究最多的方法。这类方法大部分是从针对不同分布的离群点检验方法发展起来的，通常使用分布来拟合数据集，假定所给定的数据集存在一个分布或概率模型（如正态分布或泊松分布），然后将与模型不一致（即分布不符合）的数据标识为离群数据（一般是概率分布模型具有低概率的值）。

概率分布模型通过估计用户指定的分布参数，由数据创建。如假定数据具有正态分布，则其分布的均值和标准差可以通过计算数据的均值和标准差来估计（即从训练集中估计），然后可以估计每个对象在该分布下的概率。

概率分布最常用的是正态分布。设属性 x 取自标准正态分布 $N(0,1)$，如果属性值 x 满足：$P(|x|\geqslant c)=\alpha$ 其中 c 是给定的常量，则 x 以概率 $1-\alpha$ 为离群点。式中 α 表示错误地将来自给定分布的值分类为离群点的概率，常用的是 0.05。

如果（正常对象的）一个感兴趣的属性的分布是具有均值 μ 和标准差 σ 的正态分布，即 $N(\mu,\sigma^2)$

分布，则可以通过变换 $z=(x-\mu)/\sigma$ 转换为标准正态分布 $N(0,1)$。通常 μ 和 σ 是未知的，可以通过样本均值和样本标准差来估计。实践中，当观测值很多时，这种估计的效果很好；另一方面，由概率统计中的大数定律可知，在大样本的情况下可以用正态分布近似其他分布。这样一种思想在质量控制图中有广泛应用，图 22.4 是质量控制示意图，中心线是观测值的预测值，$\mu\pm3\sigma$ 对应上下控制线，$\mu\pm2\sigma$ 对应上、下警告线，根据 3σ 原则，99.73%的观测值将落在 $\mu\pm3\sigma$ 区间内，仅有 0.27%的观测值落在此区间之外。

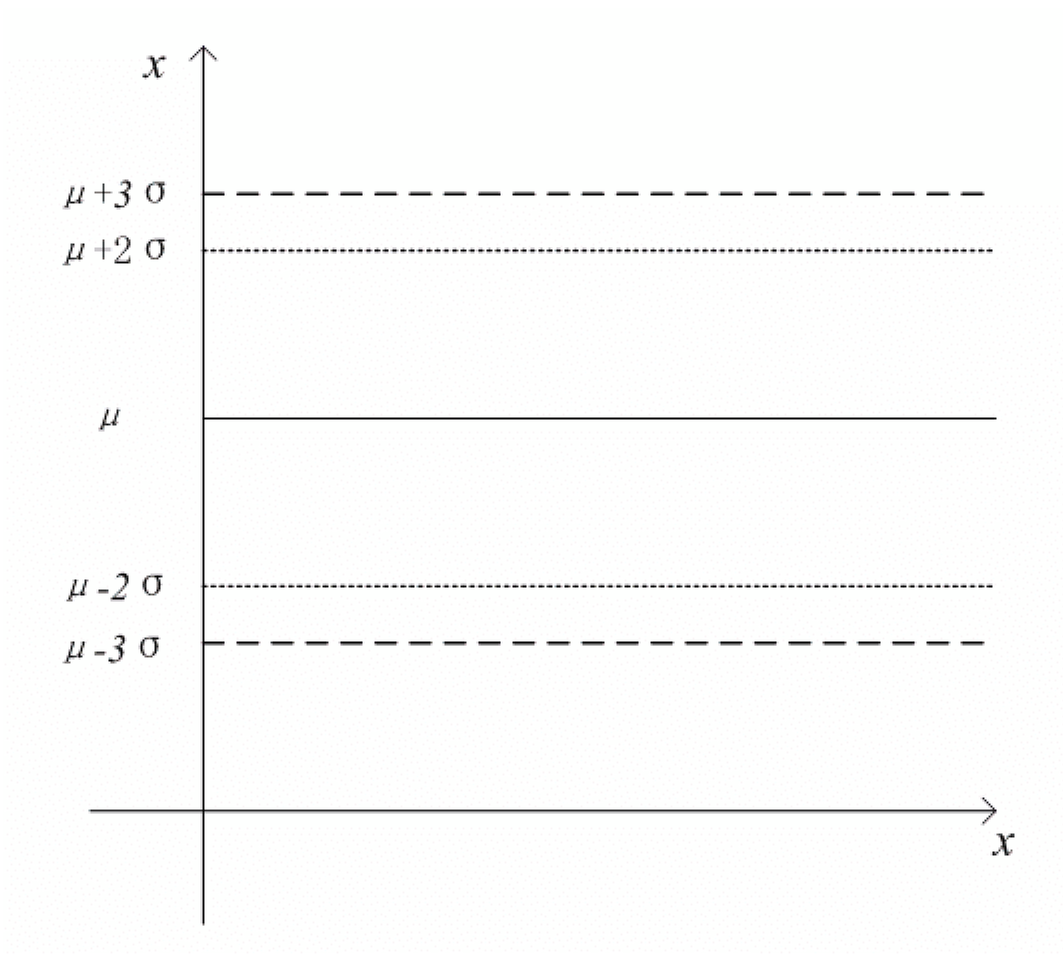


图 22.4　质量控制示意图

对于观测样本 x：

- 如果此点在上、下警告线之间区域内，则测定过程处于控制状态，生产过程或样本分析结果有效。
- 如果此点超出上、下警告线，但仍在上、下控制线之间的区域内，提示质量开始变劣，可能存在“失控”倾向，应进行初步检查，并采取相应的措施。
- 如果此点落在上、下控制线之外，表示生产或测定过程“失控”，生产的是废品或观测样本无效。应立即检查原因，予以纠正。

基于统计分布的离群点检测方法具有坚实的理论基础，当数据和所用的检验类型知识充分时，这种检验方法可能非常有效，但也存在以下几点不足。

- 尽管许多类型的数据可以用少量常见的分布（如高斯分布、泊松分布或二项式分布）来描述，但在许多应用中，数据的分布是未知的或数据几乎不可能用单一标准的分布来拟合。
- 这类方法要求已知数据集的分布类型及参数的知识。然而，在许多情况下，数据分布是未知的。当观察到的分布不能恰当地用任何标准的分布建模时，统计学方法不能确保所有的离群点被发现。另外，要确定哪种分布能最好地拟合数据集的代价也非常大。
- 这类方法绝大多数是针对低维数据的（特别是针对单个属性的），不能用于检测高维数据中的离群点。
- 这类方法不适合混合类型数据。

22.12.2　基于距离的离群点检测方法

基于距离的离群点检测方法思想直观、简单，一个对象如果远离大部分点，则认为是离群点。这种方法比统计学方法更容易使用，基于距离的离群点检测方法有多种变形，其中一种方法是利用 k-最近邻距离的大小来判定离群点的方法。

对于正整数 k，对象 p 的 k 最近邻距离 k_distance(p)定义如下。

- 除 p 外，至少有 k 个对象 o 满足 $\text{distance}(p,o)\leqslant k_\text{distance}(p)$；
- 除 p 外，至多有 $k-1$ 个对象 o 满足 $\text{distance}(p,o)<k_\text{distance}(p)$。

一个对象的最近邻的距离越大，越可能远离大部分数据，因此可以将对象的最近邻距离看成是它的离群程度（或离群点得分），称为离群因子 OF（Outlier Factor）。

点 x 的离群因子定义为

$$\mathrm{OF1}(x,k)=\frac{\sum\limits_{y\in N(x,k)}\mathrm{distance}(x,y)}{|N(x,k)|}$$

式中：$N(x,k)$是不包含 x 的 k-最近邻的集合 $N(x,k)=\{y\,|\,\mathrm{distance}(x,y)\leqslant k\text{-}\mathrm{distance}(x,y)\}$，$|N(x,k)|$是该集合的大小，其值可能大于 k。

应用此法时，需要选择合适的离群因子阈值来区分正常值和离群值。阈值可以通过图示法确定，即将 OF(x,k)降序排列，选择 OF(x,k)急速下降的点作为离群值、正常值的分隔点，如图 22.5 所示。在该图中，有两个点可判定为离群点。

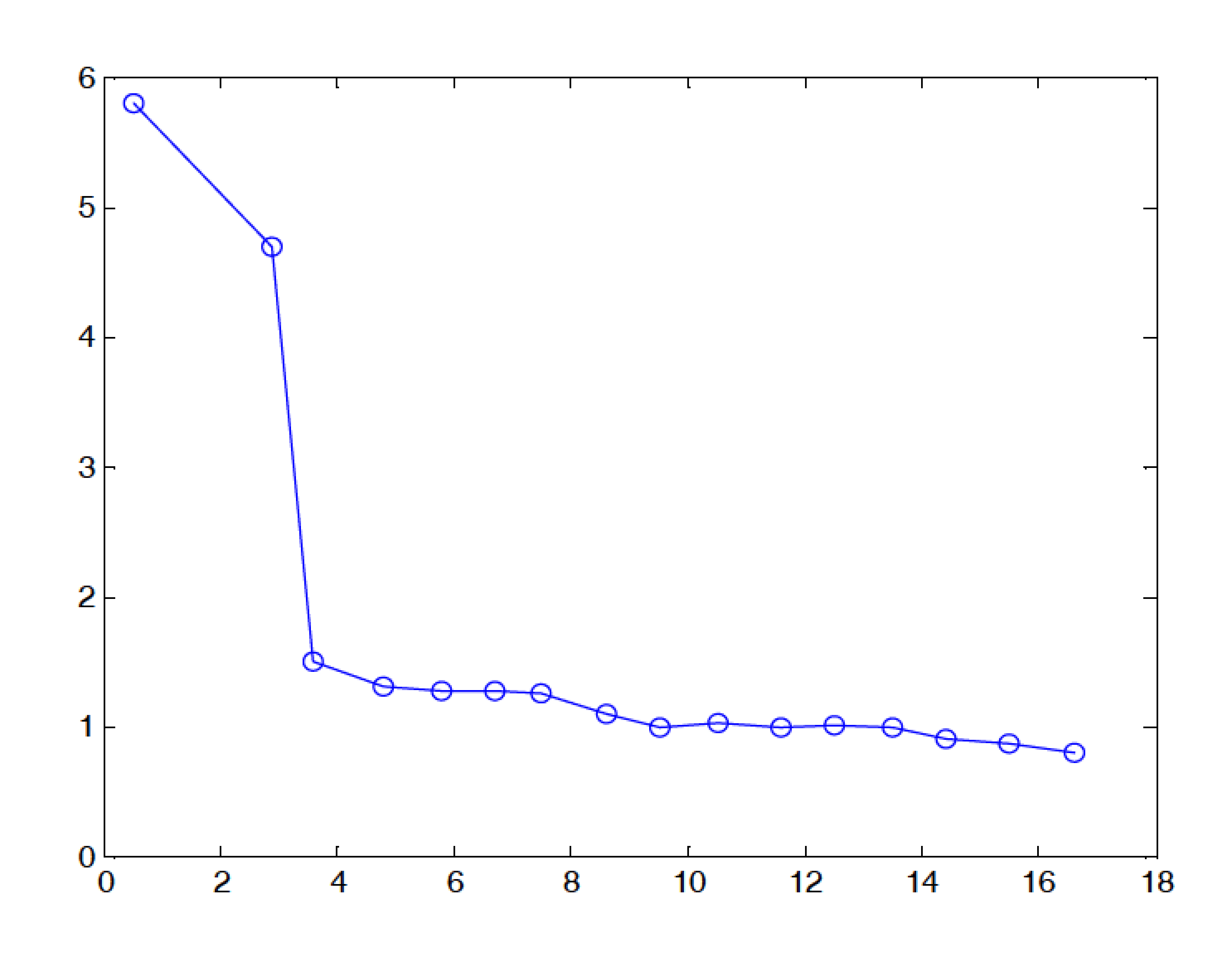


图 22.5　离群阈值选择策略示意图

基于距离的离群点检测方法简单，但该算法存在以下不足：①对 k 值较为敏感，如果 k 太小（如等于 1），则少量的邻近离群点可能导致较低的离群程度；如果 k 太大，则当点数少于 k 时，有较多的点被划分为离群点。现在还没有一种有效的方法来确定合适的值；②算法的时间复杂度为 $O(n^2)$，难以用于大规模数据集；③该算法需要有关离群因子阈值或数据集中离群点个数的先验知识，因此，在实际应用中有时由于先验知识的不足会造成一定的困难；④它要使用全局阈值，不能处理不同密度区的数据集。

22.12.3　基于相对密度的离群点检测方法

基于统计的方法与基于距离的方法都是从全局角度来考虑的全局一致的方法，不能处理不同密度区域的数据集，然而，实际应用中数据通常并非是单一分布的。当数据集含有多种分布或数据集由不同密度子集混合而成时，这些全局方法效果不佳。一个对象是否为离群点不仅仅取决于它与周围数据的距离大小，而且与邻域内的密度状况有关。一个对象的邻域密度可以用包含固定节点个数的邻域半径或指定半径邻域中包含的节点数来描述，包含固定节点数的邻域半径越大，其密度就越小；固定半径的邻域包含的节点数越多，密度就越大，因而产生了两类不同的基于密度的离群点检测方法。在此只介绍基于相对密度的离群点检测方法。

$$\mathrm{OF1}(x,k)=\frac{\sum_{y\in N(x,k)}\mathrm{distance}(x,y)}{|N(x,k)|}$$

式中：$N(x,k)$是不包含 x 的 k-最近邻的集合 $N(x,k)=\{y\mid \mathrm{distance}(x,y)\leqslant k\text{-}\mathrm{distance}(x,y)\}$，$|N(x,k)|$是该集合的大小，其值可能大于 k。

应用此法时，需要选择合适的离群因子阈值来区分正常值和离群值。阈值可以通过图示法确定，即将 OF(x,k)降序排列，选择 OF(x,k)急速下降的点作为离群值、正常值的分隔点，如图 22.5 所示。在该图中，有两个点可判定为离群点。

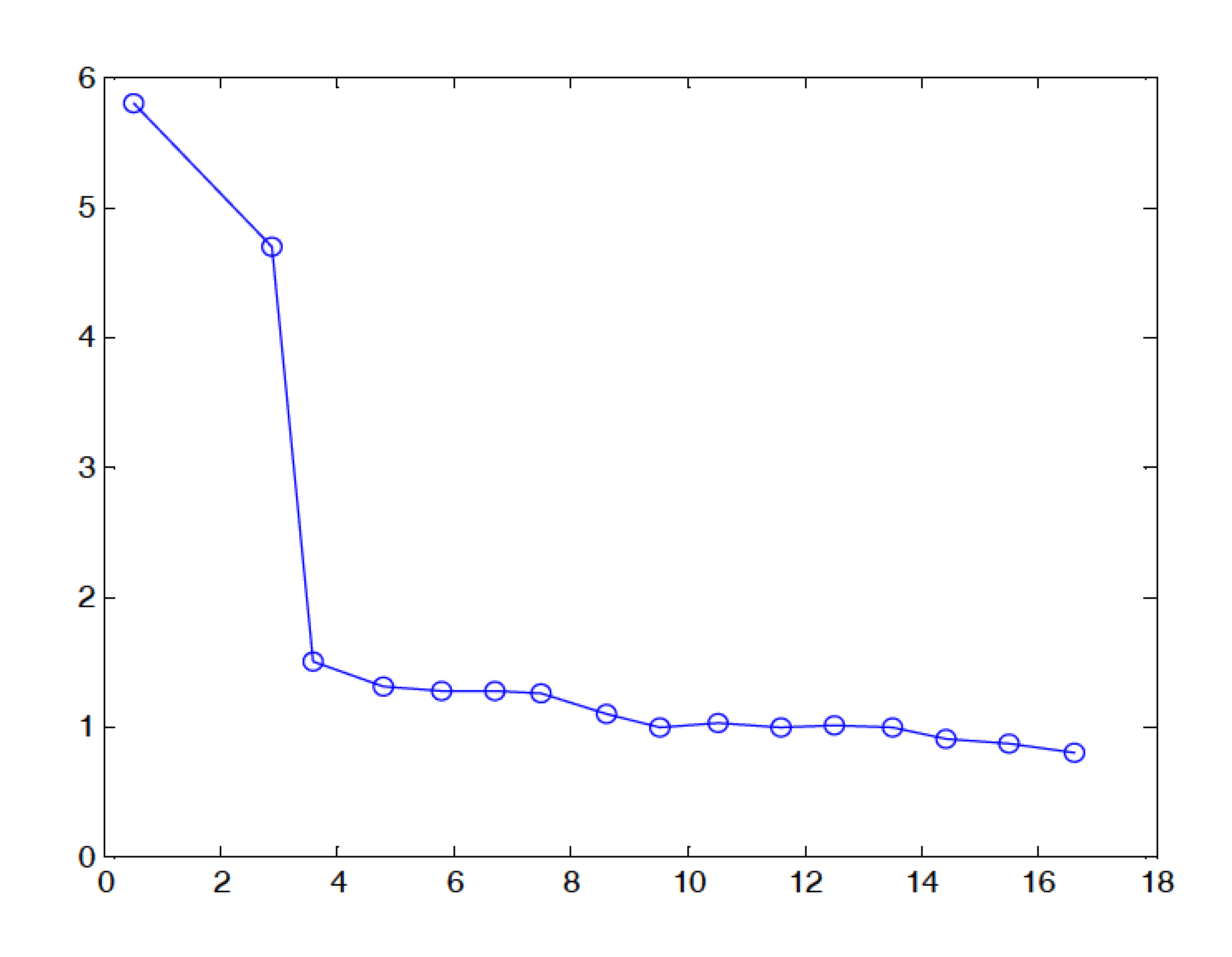


图 22.5 离群阈值选择策略示意图

基于距离的离群点检测方法简单，但该算法存在以下不足：①对 k 值较为敏感，如果 k 太小（如等于 1），则少量的邻近离群点可能导致较低的离群程度；如果 k 太大，则当点数少于 k 时，有较多的点被划分为离群点。现在还没有一种有效的方法来确定合适的值；②算法的时间复杂度为 $O(n^2)$，难以用于大规模数据集；③该算法需要有关离群因子阈值或数据集中离群点个数的先验知识，因此，在实际应用中有时由于先验知识的不足会造成一定的困难；④它要使用全局阈值，不能处理不同密度区的数据集。

22.12.3 基于相对密度的离群点检测方法

基于统计的方法与基于距离的方法都是从全局角度来考虑的全局一致的方法，不能处理不同密度区域的数据集，然而，实际应用中数据通常并非是单一分布的。当数据集含有多种分布或数据集由不同密度子集混合而成时，这些全局方法效果不佳。一个对象是否为离群点不仅仅取决于它与周围数据的距离大小，而且与邻域内的密度状况有关。一个对象的邻域密度可以用包含固定节点个数的邻域半径或指定半径邻域中包含的节点数来描述，包含固定节点数的邻域半径越大，其密度就越小；固定半径的邻域包含的节点数越多，密度就越大，因而产生了两类不同的基于密度的离群点检测方法。在此只介绍基于相对密度的离群点检测方法。

相对密度的定义如下

$$\text{relative}\quad \text{density}(x,k)=\frac{\sum_{y\in N(x,k)}\text{density}(y,k)/|N(x,k)|}{\text{density}(x,k)}$$

式中：$N(x,k)$是不包含 x 的 k-最近邻的集合；$|N(x,k)|$是该集合的大小，其值可能大于 k；density(x,k)为对象的局部邻域密度，其定义为

$$\text{density}(x,k)=\left(\frac{\sum_{y\in N(x,k)}\text{distance}(x,y)}{|N(x,k)|}\right)^{-1}$$

distance(x,y)为最近邻距离。

一个对象的 k-最近邻的距离越大，或邻域密度越小，它越可能远离大部分数据。一个数据集由多个自然簇构成，在簇内靠近核心点的对象相对密度接近于 1。而处于簇的边缘或是簇的外面的对象的相对密度相对较大，这个相对密度表示 x 是否在比它的近邻更稠密或更稀疏的邻域内，以相对密度作为 x 的离群因子，即 OF2(x,k)=relative density(x,k)，其值越大，越有可能是离群点。

基于相对密度的离群点检测算法有以下 3 个步骤。

（1）对于每一对象 x，确定 x 的 k-最近邻集合 $N(x,k)$和密度 density(x,k)。

（2）对于每一对象 x，确定 x 的相对密度 relative density(x,k)，并赋值给 OF2(x,k)。

（3）对 OF2(x,k)降序排列，确定离群因子大的若干对象。

22.12.4 基于聚类的离群点检测方法

类似于基于相对密度的方法，基于聚类的离群点检测方法也考虑到了数据的局部特性，这些方法大多利用了距离或相似度的基本概念，并通过对象或簇的特定“离群因子”来度量对象的偏离程度。

基于聚类的方法有两个共同特点：一是先采用特殊的聚类算法处理输入数据而得到簇，再在聚类的基础上来检测离群点；二是只需要扫描数据集若干次，效率较高，适用于大规模数据集。

基于聚类的离群点检测方法分为静态数据的离群点检测和动态数据的离群点的检测。静态数据的离群点检测用于离线数据的分析如税务稽查；而动态数据离群点检测用于实时性高的数据处理问题中，如在线的入侵检测。

1）静态数据的离群点检测步骤

① 对数据进行聚类，将数据划分为不相交的簇。

② 计算对象或簇的离群因子，将离群因子大的对象或簇中对象判定为离群点。

2）动态数据的离群点检测步骤

① 利用静态数据的离群点检测方法建立离群点检测模型。

② 利用对象与已有模型间的相似程度来检测离群点。

1. 基于对象的离群因子方法

这类方法的基本思路是首先聚类所有对象，然后用对象到各个簇中心的距离来度量对象偏离整

个数据集的程度。如果一个对象不强属于任何簇，是称该对象为基于聚类的离群点。由于聚类算法产生的簇的质量对该算法产生的离群点的质量有非常大的影响，因而需要选择合适的聚类算法。

该算法的检测步骤如下。

（1）对数据集 D 采用合适的聚类算法（如一趟聚类算法）进行聚类，得到聚类结果 $D=\{C_1,C_2,\cdots,C_k\}$。

（2）计算数据集 D 中所有对象的离群因子 OF3(p)，对 OF3(p)降序排列，确定离群因子大的若干对象为离群点。

在此 OF3(p)的计算公式如下

$$\mathrm{OF3}(p)=\sum_{j=1}^{k}\frac{|C_j|}{|D|}\cdot d(p,C_j)$$

式中：$|D|$为数据集 D 的大小；$|C_j|$为簇 j 的大小；$d(p,C_j)$为对象 p 到簇 j 中心的距离。

OF3(p)度量了对象 p 偏离整个数据集的程度，其值越大，说明 p 偏离整体越远，有可能是离群点。

在大样本情况下，可以将 OF3(p)近似地看成正态分布，则计算出离群因子后，然后将满足 OF3(p)≥Ave_OF+βDev_OF($1\leqslant\beta\leqslant2$)的对象 p 判定为离群点。在这里 Ave_OF 为离群因子的平均值，Dev_OF 则为其的标准差。

该算法依赖于阈值 β，其值越小，离群点的检测率越高，但误报率也会越高，通常取 $\beta=1$ 或 1.285。

2. 基于簇的离群因子检测方法

基于下面的考虑：（1）在某种度量下，相似对象或相同类型的对象会聚集在一起，或者说正常数据与离群数据会聚集在不同的簇中；（2）正常数据占绝大部分，且离群数据与正常数据的表现明显不同，或者说离群数据会偏离正常数据即大部分数据。由此可得到基于聚类的离群点检测方法：

（1）对数据集 D 采用合适的聚类算法（如一趟聚类算法）进行聚类，得到聚类结果 $D=\{C_1,C_2,\cdots,C_k\}$；

（2）计算每个簇 C_i($1\leqslant i\leqslant k$)的离群因子 OF4(C_i)，对 OF4(C_i)降序排列 C_i，求满足：$\frac{\sum_{i=1}^{b}|Ci|}{|D|}\geqslant\varepsilon\ (0<\varepsilon<1)$ 的最小 b，将簇 $C_1,C_2,\cdots,C_b$标识为 outlier 类即将每个对象均看成离群点，而将 $C_{b+1},C_{b+2},\cdots,C_k$标识为 normal 类即其中每个对象均看成正常的。

在此离群因子的定义为

$$\mathrm{OF4}(C_i)=\sum_{j=1,j\neq i}^{k}\frac{|Cj|}{|D|}\cdot d(C_i,C_j)$$

或者：
$$\text{OF5}(C_i)=\frac{k-1}{\sum_{j=1,j\neq i}^{k}\frac{1}{d(C_i,C_j)}}$$

参数 ε 对检测结果有影响。ε 实际上是离群数据所点比例的近似值，ε 越小，检测率越低，同时误报率也越低。根据统计经验，一个数据集中受污染的数据即离群数据通常小于 5%，最多不超过 15%，因此在没有先验知识的情况下一般取 ε 在 0.05～0.1 之间。实际使用时可根据性能要求和离群数据所占比例的先验知识更准确地选择。

3. 基于聚类的动态数据离群点检测

在实际领域中，很多场合离群数据的判定不是离线静态的，而是在线动态的。对于动态数据的离群检测与分类方法类似，其基本思想是：在对训练集聚类的基础上，按照簇的离群因子排序簇，并按一定比例将簇标识为 normal 或 outlier，以标识的簇作为分类模型，按照对象与分类模型中最接近簇的距离来判断它是否为离群点。具体步骤如下。

（1）模型建立。

第一步，聚类：对训练集 T_1 采用合适的聚类算法（如一趟聚类算法）进行聚类，得到聚类结果 $T_1=\{C_1,C_2,\cdots,C_k\}$；

第二步，给簇作标记：计算每个簇 $C_i(1\leqslant i\leqslant k)$的离群因子 $\text{OF}(C_i)$，对 $\text{OF}(C_i)$降序排列 C_i，求满足：$\frac{\sum_{i=1}^{b}|Ci|}{|T_1|}\geqslant\varepsilon\quad(0<\varepsilon<1)$ 的最小 b，将簇 $C_1,C_2,\cdots,C_b$ 标识为 outlier 类即将每个对象均看成离群点，而将 $C_{b+1},C_{b+2},\cdots,C_k$ 标识为 normal 类即其中每个对象均看成正常的。

第三步，确定模型：以每个簇的摘要信息，聚类半径阈值 r 确定分类模型。

（2）模型评估。

利用改进的最近邻分类评估测试集 T_2 中的每个对象，算法的具体步骤如下。

对于测试集 T_2 中对象 p，计算 p 与每个簇的距离 $d(p,C_i)$。

若：$\min\{d(p,C_i),1\leqslant i\leqslant k\}=d(p,C_{i_0})\leqslant d$

则表明 p 是已知类型的行为，可将簇 C_{i_0} 的标识作为 p 的标识，否则表明 p 是一种新的行为，将标识为可疑对象—候选离群点。

（3）模型更新。

对于测试集 T_3 中对象 p，按照前面聚类的方式，对新增对象进行增量式聚类，更新 $\overline{T}=\{\overline{C_1},\overline{C_2},\cdots,\overline{C_k}\}$ 并用建立模型同样的方法对所有簇重新标记其类别。

22.12.5 离群点挖掘方法的评估

可以通过表所示的混淆矩阵来描述离群点挖掘方法的检测性能。在离群点检测问题中，并不关注预测正确的 normal 类对象，重点关注的是正确预测的 outlier 类对象。

由于在离群点检测问题中，离群数据所占比例通常在 5%以下，常用的分类准确率的度量指

标不适合于评价离群点检测方法、检测率、误报率是度量离群点检测方法准确性的两个常用指标。检测率表示被正确检测的离群点记录数占整个离群点记录数的比例，误报率表示正常记录被检测为离群点记录数占整个正常记录数的比例。期望离群点检测方法对离群数据有高的检测率，对正常数据有低的误报率，但两个指标之间会有一些冲突，高的检测率常常会导致高的误报率。

22.13 聚类有效性

当数据不能图形化描述时，一般很难甚至不可能确定一个数据划分是否正确，这时有一定的方法来判断聚类的有效性。聚类有效性问题是一个通用问题，它涉及一个聚类算法的基本假定（簇形状、簇数目等）等是否满足等聚类的数据集，所得到的聚类结果是否能满足要求。

一个聚类过程的质量依赖于多个因素，例如初始化的方法，簇数目的选择、聚类方法等。一个好的聚类方法产生高质量的簇，即簇内的对象具有高相似度和不同簇之间具有低的相似度。

评估聚类质量的准则有两个：内部质量评价准则和外部质量评价准则。

假设数据集 D 被聚集为 k 个簇 $D=\{C_1,C_2,\cdots,C_k\}$，用 $n(C_i)$（或$|C_i|$）或 n_i 表示簇 C_i 中包含的对象个数，$n(T_j,C_i)$表示簇 C_i 中包含类别 T_j 的对象个数，则

$$n_i = n(C_i) = \sum_i n(T_j, C_i)$$

$$N = \sum_{i=1}^{k} n(C_i)$$

用 $u_{i,j}$ 表示第 i 个对象属于第 j 个簇 C_j 的隶属度，$\|\cdots\|$表示某种距离的计算。

22.13.1 内部质量评价准则

内部质量评价准则是利用有数据集的固有特征和量值来评价一个聚类算法的结果，数据集的结构未知。通常计算簇内部平均相似度、簇间平均相似度或整体相似度来评价聚类结果，内部质量评价准则与聚类算法有关，主要通过簇内距离和簇外距离的某种形式的比值来衡量，常用的包括 DB 指标、Dunn 指标、I 指标、CH 指标、Xie-Beni 指标等。

1. CH 指标

其定义为

$$V_{CH}(k) = \frac{\text{trace}B/(k-1)}{\text{trace}W/(N-k)}$$

其中：$\text{traceB} = \sum_{j=1}^{k} n_j \| z_j - z \|^2$，$\text{traceW} = \sum_{j=1}^{k} \sum_{xi \in Z_k} \| x_j - z_j \|^2$，$z$ 是整个数据集的均值，z_j 是第 j 簇的均值。CH 指标计算族间距离和簇内距离的比值，值越大，聚类结果越好。

2. *I* 指标

其定义为

$$V_I(k)=\left(\frac{E_1\times D_k}{k\times E_k}\right)^p=\left(\frac{1}{k}\times\frac{E_1}{E_k}\times D_k\right)^p$$

其中：$D_k=\max_{i,j=1}^{k}\|z_i-z_j\|$，$E_k=\sum_{j=1}^{k}\sum_{i=1}^{N}u_{ij}\|x_i-z_j\|$，$p$ 用来调整不同的簇结构的对比，通常取 2。使聚类有效性函数 $I(k)$ 最大的值，就是最优的簇个数。

3. Xie-Beni 指标

此指标是数据集和簇中心的函数，它是紧密度和分离度的一个比值，其定义为

$$V_{XB}(k)=\frac{\sum_{j=1}^{k}\sum_{i=1}^{N}u_{ij}^2\|x_i-z_j\|^2}{N\bullet\min_{i,j}\{\|z_i-z_j\|^2\}}$$

一个好的聚类结果应使它的输入样本尽可能靠近它们的簇的中心，且使所有的簇中心尽可能远离。因此，*XB* 的定义中，对象与簇中心距离加权和的平均值表示簇内的平均紧密度，紧密度越大，则该平均加权和越小；簇中心之间的最小距离表示簇的分离度，因此簇的紧密度越大、分离度越大，则该度量值越小。即一个较小的 *XB* 度量值对应着一个较好的聚类结果。

4. Davies-Bouldin 指标

其定义为

$$V_{DB}(k)=\frac{\sum_{j=1}^{k}\max_{j,j\neq i}\left\{\frac{S_i+S_j}{d_{ij}}\right\}}{k}$$

其中：$S_i=\frac{1}{n_i}\sum_{x\in C_i}^{k}\|x-z_i\|$度量了 C_i 簇的样本之间的紧密程度，$di_j=\|z_i-z_j\|$度量簇 C_i 的样本与簇 C_j 的样本之间的分散程度。DB 指标实际上是关于同一类中样本的紧密程度与不同簇之间样本分散程度的一个函数，从几何学的角度，使簇内样本间距最小而簇间样本距离最大的分类应该是最佳的分类结果，因此，使 DB 最小化的类别数就是最优类别数。

5. Dunn 指标

从几何学的角度看，指标与 DB 指标的基本原理是相同的，它们都适用于处理簇内样本分布紧密、而簇间样本分布分散的数据集合。设 S 和 T 是非空数据集，S 的直径 $\varDelta$，S 与 T 之间的距离 δ 分别定义为

$$\varDelta(S)=\max_{x,y\in S}\{d(x,y)\}\qquad\qquad \delta(S,T)=\max_{x\in S,y\in T}\{d(x,y)\}$$

其中：$d(x,y)$表示两个对象间的距离。

Dunn 的有效性指标定义为

$$V_D(k)=\min_{1\leqslant i\leqslant c}\left\{\min_{\substack{1\leqslant j\leqslant c\\ j\neq i}}\left\{\frac{\delta(C_i,C_j)}{\max_{1\leqslant i\leqslant k}\{\Delta(C_l)\}}\right\}\right\}$$

使 V_D 最大的类别数 k，即为最佳类别数。

在以上这些指标中，研究表明，I 指标与 CH 指标的效果相对较好。

22.13.2　外部质量评价准则

外部质量评价准则是基于一个已经存在的人工分类数据集（已知每个对象的类别）进行评价的，这样可以将聚类输出结果直接与之进行比较。外部质量评价准则与聚类算法无关，理想的聚类结果是：具有相同类别的数据被聚集到相同的簇中，具有不同类别的数据聚集在不同的簇中。

可以采用聚类熵作为外部质量的评价准则。考虑簇中不同类别数据的分布，对于簇 C_i，聚类熵的定义为

$$e(C_i)=-\sum_j\frac{n(T_j,C_i)}{n(C_i)}\log\frac{n(T_j,C_i)}{n(C_i)}$$

整体聚类熵定义为所有聚类熵的加权平均值

$$e=\frac{1}{\sum_{i=1}^{m}n(C_i)}\sum_{i=1}^{m}n(C_i)e(C_i)$$

聚类熵越小，聚类效果越好。

评估聚类结果质量的另一外部质量评价准则为聚类精度，基本出发点是使用簇中数目最多的类别作为该簇的类别标记。对于簇 C_i，聚类精度的 $\phi(C_i)$定义为

$$\phi(C_i)=\frac{1}{n(C_i)}\max_j\{n(T_j,C_i)\}$$

整体聚类精度定义为所有聚类精度的加权平均值

$$\phi=\frac{1}{\sum_{i=1}^{k}n(C_i)}\sum_{i=1}^{k}n(C_i)\phi(C_i)=\frac{\sum_{i=1}^{k}N_i}{N}$$

其中：$N_i=\max_j\{n(T_j,C_i)\}$ 是簇 C_i 中占支配地位的类别的对象数，$1-\phi$ 定义为相对聚类错误率。聚类精度 ϕ 大或聚类错误率 $1-\phi$ 小，说明聚类算法将不同类别的记录较好地聚集到了不同的簇中，其聚类准确性高。

22.14　例题

下面介绍常见的聚类方法，这些方法可以解决绝大多数的聚类问题。大数据的聚类方法可参见相关资料。

例 4.37　对选定的秦川牛、晋南牛、南阳牛、延边牛、复州牛、鲁西牛和郏县红牛 7 个良种黄牛品种，可以用 15 个性能指标衡量（如表 22.4 所示）。请用此系统聚类法分类。

表 22.4　7 个黄牛品种 15 个性能指标

	秦 川 牛	晋 南 牛	南 阳 牛	延 边 牛	复 州 牛	鲁 西 牛	郏县红牛
x_1	0.8375	0.7931	0.6625	0.9114	0.7015	0.8125	0.8500
x_2	0.1000	0.1379	0.0500	0.0886	0.2463	0.0833	0.0500
x_3	0.0625	0.0690	0.2878	0.0000	0.0522	0.0938	0.1000
x_4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0104	0.0000
x_5	0.5500	0.5667	0.5250	0.5942	0.7891	0.7128	0.6667
x_6	0.4500	0.4333	0.4750	0.3841	0.2110	0.2447	0.3205
x_7	0.0000	0.0000	0.0000	0.0217	0.0000	0.0000	0.0000
x_8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0426	0.0123
x_9	0.1396	0.1380	0.0732	0.3223	0.1797	0.1023	0.0769
x_{10}	0.5466	0.7241	0.7195	0.4079	0.6172	0.3750	0.3589
x_{11}	0.0000	0.0000	0.0000	0.0066	0.0078	0.3525	0.2821
x_{12}	0.0233	0.0345	0.0122	0.0000	0.0000	0.0000	0.0000
x_{13}	0.2907	0.1035	0.1951	0.1842	0.9153	0.1705	0.2821
x_{14}	0.1938	0.1982	0.1736	0.6784	0.0157	0.0774	0.0513
x_{15}	0.8061	0.8062	0.8264	0.9216	0.9843	0.9226	0.9487

解：

设定阈值 0.4，利用最小距离法可对 7 类分成：

```
>>load mydata;
>>L=pdist(x);L_min=min(L);L_max=max(L);              %样本间距离范围，从而确定阈值
>> y=syscluster(x,0.4,'single')
 y =1    1    1    2    3    4    4                   %分成 4 类
>>y=clusterdata(x,'linkage','single','maxclust',4);  %MATLAB 自带的系统聚类函数
 y=2    2    2    3    4    1    1
```

两者分类结果完全一致。

```
 >> y=clusterdata(x,'linkage','single','cutoff',0.4);  %分成了 5 类
 y=5    5    1    3    4    2    2
```

例 4.38　某地区内有 12 个气象观测站，10 年来各站测得的年降水量如表 22.5 所示。为了节省开支，想要适当减少气象观测站，试问减少哪些观察站可以使所得到的降水量信息仍然足够大?

表 22.5　年降水量　　单位：mm

站　点	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990
1	276.2	251.5	192.7	246.2	291.7	466.5	258.6	453.4	158.2	324.8
2	324.5	287.3	433.2	232.4	311.0	158.9	327.4	365.5	271.0	406.5
3	158.6	349.5	289.9	243.7	502.4	223.5	432.1	357.6	410.2	235.7
4	412.5	297.4	366.3	372.5	254.0	425.1	403.9	258.1	344.2	288.8

续表

站　点	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990
5	292.8	227.8	466.2	460.4	245.6	251.4	256.6	278.8	250.0	192.6
6	258.4	453.6	239.1	158.9	324.8	321.0	282.9	467.2	360.7	284.9
7	334.1	321.5	357.4	298.7	401.0	315.4	389.7	355.2	376.4	290.5
8	303.2	451.0	219.7	314.5	266.5	317.4	413.2	228.5	179.4	343.7
9	292.9	466.2	245.7	256.6	251.3	246.2	466.5	453.6	159.2	283.4
10	243.2	307.5	411.1	327.0	289.9	277.5	199.3	315.6	342.4	281.2
11	159.7	421.1	357.0	296.5	255.4	304.2	282.1	456.3	331.2	243.7
12	331.2	455.1	353.2	423.0	362.1	410.7	387.6	407.2	377.7	411.1

解：

对于这个问题可以利用已有的12个气象观测站的数据进行模糊聚类分析，最后确定从哪几类中去掉几个观测站。

（1）建立模糊集合。

设A_j表示第j个观测站的降水量信息，则其隶属度函数为

$$\mu_{A_j}(x)=\mathrm{e}^{-\left(\frac{x-a_j}{b_j}\right)^2}$$

其中：a_j为每个观察站十年间观察值的平均值：$a_j=\frac{\sum_{i=1}^{10}a_{ij}}{10}$，$b_j$为其标准差：$b_j=\sqrt{\frac{1}{9}\sum_{i=1}^{10}(a_{ij}-a_j)^2}$

（2）利用格贴近度建立模糊相似矩阵：

令

$$r_{ij}=\mathrm{e}^{-\left(\frac{a_j-a_i}{b_i+b_j}\right)^2}$$

求得模糊相似矩阵 $R=(r_{ij})_{12\times12}$。

（3）求R的传递闭包。

求得 R^4 是传递闭包，也就是所求的等价矩阵。取 λ=0.998，把观测站分为合适的类。再对各类的情况进行分析，就可以得出可以撤销哪几个站。

根据以上的步骤，编程进行计算，可得到观察站可以分成以下4类：

$$\{x_1,x_5\}、\{x_2,x_3,x_6,x_8,x_9,x_{10},x_{11}\}、\{x_4,x_7\}、\{x_{12}\}$$

上述分类具有明显的意义，$\{x_1,x_5\}$属于该地区10 年中平均降水量偏低的观测站，$\{x_4,x_7\}$属于该地区10年中平均降水量偏高的观测站，$\{x_{12}\}$是平均降水量最大的观测站，而其余观测站属于中间水平。

显然，去掉的观测站越少，则保留的信息量越大。为此，考虑在去掉的观测站数目确定的条件下使得信息量最大的准则。由于该地区的观测站分为4 类，且第4类只含有一个观测站，因此，

续表

站　点	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990
5	292.8	227.8	466.2	460.4	245.6	251.4	256.6	278.8	250.0	192.6
6	258.4	453.6	239.1	158.9	324.8	321.0	282.9	467.2	360.7	284.9
7	334.1	321.5	357.4	298.7	401.0	315.4	389.7	355.2	376.4	290.5
8	303.2	451.0	219.7	314.5	266.5	317.4	413.2	228.5	179.4	343.7
9	292.9	466.2	245.7	256.6	251.3	246.2	466.5	453.6	159.2	283.4
10	243.2	307.5	411.1	327.0	289.9	277.5	199.3	315.6	342.4	281.2
11	159.7	421.1	357.0	296.5	255.4	304.2	282.1	456.3	331.2	243.7
12	331.2	455.1	353.2	423.0	362.1	410.7	387.6	407.2	377.7	411.1

解：

对于这个问题可以利用已有的12个气象观测站的数据进行模糊聚类分析，最后确定从哪几类中去掉几个观测站。

（1）建立模糊集合。

设A_j表示第j个观测站的降水量信息，则其隶属度函数为

$$\mu_{A_j}(x)=\mathrm{e}^{-\left(\frac{x-a_j}{b_j}\right)^2}$$

其中：a_j为每个观察站十年间观察值的平均值：$a_j=\frac{\sum_{i=1}^{10}a_{ij}}{10}$，$b_j$为其标准差：$b_j=\sqrt{\frac{1}{9}\sum_{i=1}^{10}(a_{ij}-a_j)^2}$

（2）利用格贴近度建立模糊相似矩阵：

令

$$r_{ij}=\mathrm{e}^{-\left(\frac{a_j-a_i}{b_i+b_j}\right)^2}$$

求得模糊相似矩阵 $R=(r_{ij})_{12\times12}$。

（3）求R的传递闭包。

求得 R^4 是传递闭包，也就是所求的等价矩阵。取 λ=0.998，把观测站分为合适的类。再对各类的情况进行分析，就可以得出可以撤销哪几个站。

根据以上的步骤，编程进行计算，可得到观察站可以分成以下4类：

$\{x_1,x_5\}$、$\{x_2,x_3,x_6,x_8,x_9,x_{10},x_{11}\}$、$\{x_4,x_7\}$、$\{x_{12}\}$

上述分类具有明显的意义，$\{x_1,x_5\}$属于该地区10 年中平均降水量偏低的观测站，$\{x_4,x_7\}$属于该地区10年中平均降水量偏高的观测站，$\{x_{12}\}$是平均降水量最大的观测站，而其余观测站属于中间水平。

显然，去掉的观测站越少，则保留的信息量越大。为此，考虑在去掉的观测站数目确定的条件下使得信息量最大的准则。由于该地区的观测站分为4 类，且第4类只含有一个观测站，因此，

通过计算下式来判断从前3类中去掉的站点

$$\min err = \sum_{i=1}^{10}(\overline{d_{i3}} - \overline{d_i})^2$$

其中：$\overline{d_i}$ 表示该地区第i年的平均降水量；$\overline{d_{i3}}$ 表示该地区去掉3个观测站以后第i年的平均降水量。通过计算可知可以去掉的观察站为｛x_7,x_9,x_{10}｝。

```
>>load mydata;
>>mu=mean(a),sigma=std(a);
>>for i=1:12
   for j=1:12
     r(i,j)=exp(-(mu(j)-mu(i))^2/(sigma(i)+sigma(j))^2);
   end
 end
>> [y,b]=fuz_eqvalue(r);                              %等价集
>> y=fuzzr(y,0.998);                                  %截集
>> y=fuz_class(y);                                    %分类
>>aa=nchoosek([1 2 3 4 5 6 7 8 9 10 11],3);          %各种方案
>>bb=size(aa,1);
>>a1=mean(a,2);
>>for i=1:bb
  a2=redu(a,aa(i,:),'c');                             %去掉3个观察站的数据集
  a3=mean(a2,2);
  err(i)=sum((a1-a3).^2);
 end
>> [a4,b1]=min(err);
>>yy=aa(b1,:)
 yy=7   9   10
>> a4
 a4=176.6312                                          %此方案的误差
```

例 4.39　对 15 个样品进行某 4 项指标的测定，结果如表 22.6 所示。可以认为它们可分为 3 类，但不知道具体哪一样本对应的类别，试对它们进行自动归类。

表 22.6　原始数据　　单位：mg/kg

序　号	x_1	x_2	x_3	x_4
1	11.853	0.480	14.360	25.210
2	45.596	0.526	13.850	24.040
3	3.525	0.086	24.400	49.300

续表

序　号	x_1	x_2	x_3	x_4
4	3.681	0.327	13.570	25.120
5	48.287	0.386	14.500	25.900
6	4.741	0.140	6.900	15.700
7	4.223	0.340	3.800	7.100
8	6.442	0.190	4.700	9.100
9	16.234	0.390	3.400	5.400
10	10.585	0.420	2.400	4.700
11	48.621	0.082	2.057	3.847
12	288.149	0.148	1.763	2.968
13	316.604	0.317	1.453	2.432
14	307.310	0.173	1.627	2.729
15	82.170	0.105	1.217	2.188

解：

利用遗传算法进行分类，可以用两种方法。一种是对分类中心操作，即先随机生成分类中心，然后再计算各样品与各类中心的距离，并将其归入距离最小的类别中，然后对类中心优化，最终得到各样品的分类。这一种方法可用于所有现代优化方法，如粒子群算法、蚁群算法、鱼群算法等；另一种方法是利用形如以下的编码：（1 2 3 3 3 2 1 3 3 3 3 3 2 2 1），其中编码长度为样本数量，序号即为样本序号，数字对应该样本的类别。上述编码表明序号为1、7和15的样本为第1类，第2、6、13、14号的样本为第2类，第3、4、5、8、9、10、11、12号的样本为第3类。然后对编码进行操作，最终得到较优的分类。这种方法需要自己对遗传算法编码。

```
>>load data;
>>y=gaclustering(x,3,50,800);                  %第1种方法
 y=1  1  1  1  1  2   2  2  2  2  2  3  3  3  2
>> [y,k]=ga_cluster(x,3,50,[],[],1,800)     %第2种方法
 y=[1x6 double]    [1x8 double]    [3]
 y{1}=7   8  11  12  14  15
 y{2}=1  2  4   5   6   9   10   13
```

这两种方法得到的结果有所差异，这主要是因适应度函数不同而造成的，另外遗传算法的随机性也对结果有一定的影响。

例 4.40　为了解耕地的污染状况与水平，从 3 块由不同水质灌溉的农田里共取 16 个样品，每个样品均作土壤中铜、镉、氟、锌、汞和硫化物等 7 个变量的浓度分析，原始数据如表 22.7 所示。试用蚁群算法对 16 个样品进行分类。

表 22.7　原始数据　　单位：mg/kg

序　号	x_1	x_2	x_3	x_4
1	11.853	0.480	14.360	25.210
2	3.681	0.327	13.570	25.120
3	48.287	0.386	14.500	25.900
4	4.741	0.140	6.900	15.700
5	4.223	0.340	3.800	7.100
6	6.442	0.190	4.700	9.100
7	16.234	0.390	3.400	5.400
8	10.585	0.420	2.400	4.700
9	48.621	0.082	2.057	3.847
10	288.149	0.148	1.763	2.968
11	316.604	0.317	1.453	2.432
12	307.310	0.173	1.627	2.729
13	82.170	0.105	1.217	2.188
14	3.777	0.870	15.400	28.200
15	62.856	0.340	5.200	9.000
16	3.299	0.180	3.000	

解：

首先通过 MATLAB 中的聚类函数，求出样品间的聚类情况。当用最小距离法时，样品间的聚类树如图 22.6 所示。可见根据不同的标准，可以有多种划分方法。

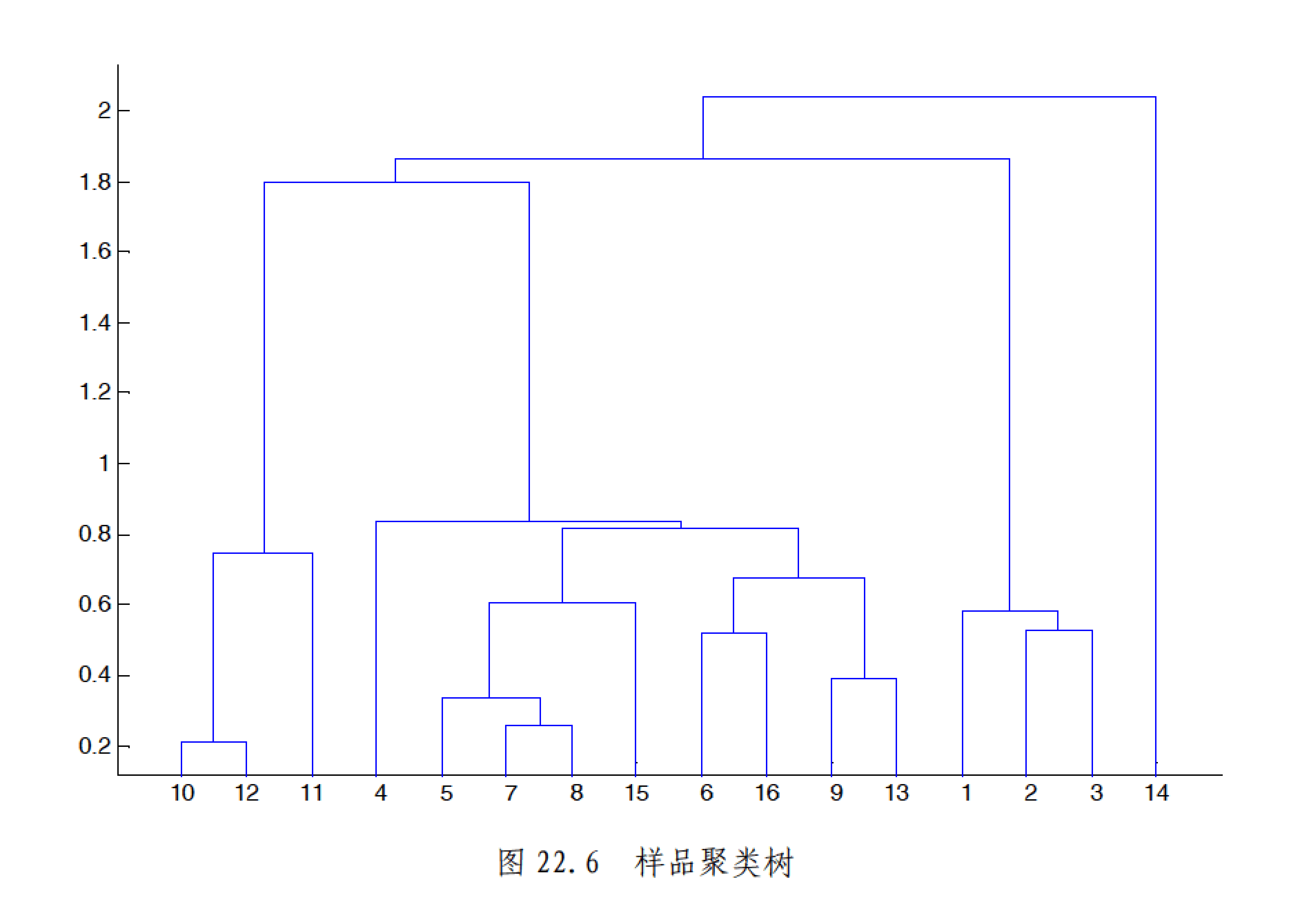


图 22.6　样品聚类树

为了简单起见，本例用蚁群算法聚类时分为 3 类。

根据蚁群算法原理，设计 17 层城市，其中除了前后两座城市，其余各层均为 3 个城市，代表类别数。每只蚂蚁从左到右所找到的路径即代表各样品所对应的类别，而每次移动的路径，则受层间信息素和各样品与类之间的信息素的共同作用。每次移动后对路径间的信息素进行局部更新。

当所有 m 只蚂蚁按上述过程完成一次循环，就对样品与各类别间的信息素进行全局更新。首先对每只蚂蚁经过的路径解码，得到各样品所对应的类别，由此计算优化函数，并得到最小值。根据函数最小值对应的路径更新样品与类别间的信息素。

根据蚁群算法的基本原理，可以编制相应的程序计算，结果得到如下的路径：1－1－1－2－2－2－2－2－2－3－3－3－4－5－6－6。

```
>>load data;
>>m_pattern=antcluster1(data,[],[],[],[]);
m_pattern =1  1  1  2  2  2  2  2  2  3  3  3  4  5  6  6
```

如果事先不知道聚类的数目，则可以根据样本间的距离矩阵，确定一个阈值距离。当多个类之间的距离小于此值时，根据概率选择其中两个类的归并，而概率大小与路径的信息素有关，规定当两类之间的距离小于阈值时，信息素为 1，否则为 0。

例 4.41　在许多问题中样品是依次排列的（如以时间、地理位置或优劣为序），在它们分类时，不能打乱样品的次序，称为有序样品的聚类，其中最常用的方法称最优分割法。例如对动植物按生长的年龄段进行分类，年龄的顺序是不能改变的，否则就没有实际意义。

为了了解儿童的生长发育规律，随机抽样统计了男孩从出生到 11 岁每年平均增长体重的重量数据如表 22.8 所示，试问男孩发育可分为几个阶段?

表 22.8　1～11 岁男孩每年平均增长的体重

年龄（岁）	1	2	3	4	5	6	7	8	9	10	11
增重（kg）	9.3	1.8	1.9	1.7	1.5	1.3	1.4	2.0	1.9	2.3	2.1

解：

设 $x_1,x_2,\cdots,x_n$ 为有序样品，希望在不改变下标的条件下将它们分成类，即

$$G_1=\{x_1,\cdots,x_{i_1}\},G_2=\{x_{i_1+1},\cdots,x_{i_2}\},\cdots,G_k=\{x_{i_{k-1}+1},\cdots,x_n\}$$

其中：$0<i_1<i_2<i_{k-1}<n$，并称 $G_1,G_2,\cdots,G_k$ 为样品 G 的一个 k 分割。对于这样的分割，共有 C_{n-1}^{k-1} 个。

对于给定的 $0<i_1<\cdots<i_{k-1}<n$，则 $i_1,\cdots,i_{k-1}$ 代表一种 k 分割，即令

$$S_n(k;i_1,\cdots,i_{k-1})=D_{0,i_1}+D_{i_1,i_2}+\cdots+D_{i_{k-1},n}$$

为对应 k 分割的总变差，式中 $D_{i,j}$ 为类 $G_{i,j}=\{x_{i+1},\cdots,x_j\}$（$i<j$）的距离。

显然，S_n 越小，各类间的距离也越小，分类也越合理。因此，只要能使

$$S_n(k;i_1,\cdots,i_{k-1})=\min_{0<j_1<\cdots<j_{k-1}<n}S_n(k;j_1,\cdots,j_{k-1})$$

便可以得到最优的分割。

最优分割可以采用穷举法，即将 C_{n-1}^{k-1} 种分割方法穷举出来，然后找到最小总变差的分割，也可以采用动态规划的方法进行求解，即 Fisher 最优求解法，下面是求解过程。

① 对于给定的有序样本集，可计算如下的距离表：

$$
\begin{array}{ccccc}
D_{0,1} D_{0,2} D_{0,3} & \cdots & D_{0,n} \\
D_{1,2} D_{1,3} & \cdots & D_{1,n} \\
D_{2,3} & \cdots & D_{2,n} \\
& \vdots & \\
& & D_{n-1,n}
\end{array}
$$

② 求最优二分割的方法。首先将有序样本作 n-1 种的二分割法，即

$\{\{\{x_1,\cdots,x_{n-1}\},\{x_n\}\};\{\{x_1,\cdots,x_{n-2}\},\{x_{n-1},x_n\}\},\cdots,\{\{x_1,x_2\},\{x_3,\cdots x_n\}\},\{\{x_1\},\{x_2,\cdots,x_n\}\}\}$

每种分法各对应一个总变差，即

$$
\begin{aligned}
&S(2,2)=D_{0,1}+D_{1,2}\\
&S(3,2)=\min(D_{0,1}+D_{1,3},D_{0,2}+D_{2,3})\\
&\vdots\\
&S(n,2)=\min_{2\leqslant j\leqslant n}\{D_{0,j}+D_{j,n}\}
\end{aligned}
$$

同时，记录最优划分的位置 $p(i,2)$，$2\leqslant i\leqslant n$。

③ 求最优三分割的方法。用类似的方法求出最优三分割、四分割、一直到 k 分割。

④ 分类个数(k)的确定。如果能从实际问题中事先确定 k 当然最好。如果不能，可以从 $S(n,k)$ 随 k 的变化趋势图中找到拐点处，作为确定 k 根据。当曲线拐点很平缓时，可选择的 k 较多，这时需要用其他的方法来确定，如均方比和特征根法。

编制相应的程序，求出 $S(n,k)$及对应的分类位置：

得到如下 S 的结果，其中 k 为 2～10，l 为 3～11，括号中的数字为 g 值，表示分类的位置。

```
s=[0.0050(2)
0.0200(2)0.0050(4)
0.0875(2)0.0200(5) 0.0050(5)
0.2320(2)0.0400(5) 0.0200(6) 0.0050(6)
0.2800(2)0.0400(5) 0.0250(6) 0.0100(6) 0.0050(6)
0.4171(2)0.2800(8) 0.0400(8) 0.0250(8) 0.0100(8) 0.0050(8)
0.4688(2)0.2850(8) 0.0450(8) 0.0300(8) 0.0150(10)0.0100(8) 0.0050(8)
0.8022(2)0.3667(8) 0.1267(8) 0.0450(10)0.0300(10)0.0150(10)0.0100(10) 0.0050(10)
0.9090(2)0.3675(8)  0.1275(8)  0.0650(10) 0.0450(11)0.0300(11)0.0150(11)0.0100(10) 0.0050(11)]
```

根据 $S(n,k)$与 k 的曲线，选 k=4，即儿童生长可分成 4 个阶段。则根据 $S(11,4)=0.1275$（8），可知最优损失函数值为 0.1275，最后的分割在第 8 个元素处，因此 G_4 包含的样本为｛8～11｝，

然后根据求 S 最小值可得 $S(5,3)=0.020(5)$，即 G_3 包含的样本为｛5～7｝，类似 S(4,2)=0.020(2)，G_2 中的样本为｛2,3,4｝，$G_1=\{1\}$。

```
>> y=order(x,4)
  y =1  2  2  2  3  3  3  4  4  4  4
```

即样本的最优有序分类为：｛9.3｝，｛1.8，1.9，1.7｝，｛1.5，1.3，1.4｝，｛2.0，1.9，2.3，2.1｝。

例 4.42　评定某一职位的任职资格时，提出了 15 个指标，即 x_1 申请书印象，x_2 学术能力，x_3 讨人喜欢，x_4 自信程度，x_5 精明，x_6 诚实，x_7 推销能力，x_8 经验，x_9 积极性，x_{10} 抱负，x_{11} 外貌，x_{12} 理解能力，x_{13} 潜力，x_{14} 交际能力，x_{15} 适应能力。但这些指标是否合适值得商榷。希望通过相应的分析，考查各指标的重要性以减少指标值。表 22.9 是 9 名考察对象 15 项指标得分情况，请对此进行分析。

表 22.9　9 名考察对象 15 项指标得分情况

观察对象 / 指标	1	2	3	4	5	6	7	8	9
X_1	6	9	7	5	6	7	9	9	9
X_2	2	5	3	8	8	7	8	9	7
X_3	5	8	6	5	8	6	8	8	8
X_4	8	10	9	6	4	8	8	9	8
X_5	7	9	8	5	4	7	8	9	8
X_6	8	9	9	9	9	10	8	8	8
X_7	8	10	7	2	2	5	8	8	5
X_8	3	5	4	8	8	9	10	10	9
X_9	8	9	9	4	5	6	8	9	8
X_{10}	9	9	9	5	5	5	10	10	9
X_{11}	7	10	8	6	8	7	9	9	9
X_{12}	7	8	8	8	8	8	8	9	8
X_{13}	5	8	6	7	8	6	9	9	8
X_{14}	7	8	8	6	7	6	8	9	8
X_{15}	10	10	10	5	7	6	10	10	10

解：

```
>>x1=[6 9 7 5 6 7 9 9 9];x2=[2 5 3 8 8 7 8 9 7];x3=[5 8 6 5 8 6 8 8 8];
>>x4=[8 10 9 6 4 8 8 9 8];x5=[7 9 8 5 4 7 8 9 8];x6=[8 9 9 9 9 10 8 8 8];
  x7=[8 10 7 2 2 5 8 8 5];>>x7=[8 10 7 2 2 5 8 8 5];x8=[3 5 4 8 8 9 10 10 9];
  x9=[8 9 9 4 5 6 8 9 8];>>x10=[9 9 9 5 5 5 10 10 9];x11=[7 10 8 6 8 7 9 9 9];
  x12=[7 8 8 8 8 8 8 9 8];>>x13=[5  8 6 7 8 6 9 9 8];x14=[7 8 8 6 7 6 8 9 8];
  x15=[10 10 10 5 7 6 10 10 10];
>> y=graycluster(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15);  %阈值默
认为 0.8
```

```
>> y=1  2  1  3  4  1  5  2  5  51  1  1  1  5      %分类结果
>> y=graycluster(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,0.58);
>>y=1  1 1  2  1  1   2  1  2  2  1  1  1  1  2
```

可以将同一类的指标并于一个统一的指标，就可以大大减少指标的数量，以减少模型的可靠性。但需要考虑的是选择的阈值，使分类比较合理。

例 4.43　利用人工鱼群算法对表 22.10 的 Iris 数据进行分类处理。

表 22.10　Iris 数据

特征 / 类别	萼片长	萼片宽	花瓣长	花瓣宽
Iris-setosa	5.0	3.4	1.6	0.4
	5.2	3.5	1.5	0.2
	5.2	3.4	1.4	0.2
	4.7	3.2	1.6	0.2
	4.8	3.1	1.6	0.2
	5.4	3.4	1.5	0.4
versicolor	7.0	3.2	4.7	1.4
	6.4	3.2	4.5	1.5
	6.9	3.1	4.9	1.5
	5.5	2.3	4.0	1.3
	6.5	2.8	4.6	1.5
	5.7	2.8	4.5	1.3
	6.3	3.3	4.7	1.6
virginica	6.5	3.0	5.8	2.2
	7.6	3.0	6.6	2.1
	4.9	2.5	4.5	1.7
	7.3	2.9	6.3	1.8
	6.7	2.5	5.8	1.8
	7.2	3.6	6.1	2.5
	6.5	3.2	5.1	2.0

解：

利用鱼群算法对数据进行聚类的过程，是对每条鱼所代表的聚类中心进行迭代寻优，最终找到最优鱼所代表的聚类中心，然后再依据此聚类中心计算每个样品与各类聚类中心的距离，最后确定类别。

```
>>x1=[5.0 3.4 1.6 0.4;5.2 3.5 1.5 0.2;5.2 3.4 1.4 0.2;4.7 3.2 1.6 0.2;4.8 3.1 1.6 0.2;
     5.4 3.4 1.5 0.4;7.0 3.2 4.7 1.4;6.4 3.2 4.5 1.5;6.9 3.1 4.9 1.5;5.5 2.3 4.0 1.3;
     6.5 2.8 4.6 1.5;5.7 2.8 4.5 1.3;6.3 3.3 4.7 1.6;6.5 3.0 5.8 2.2;7.6 3.0 6.6 2.1;
     4.9 2.5 4.5 1.7;7.3 2.9 6.3 1.8;6.7 2.5 5.8 1.8;7.2 3.6 6.1 2.5;6.5 3.2 5.1 2.0];
```

```
>>iterate_times=50;af=[0.7 3 50 8 10];x2=3;
>> [best_class,best_value]=fish4(x1,af,iterate_times,x2);
best_class =[2 2 2 2 2 2 1 1 1 1 1 1 1 3 3 1 3 3 3 3]
```

从结果看，有一个结果分错，但从这个样品的数据可以看出它分在第二类较为合适。

例 4.44　利用基于密度的聚类法对随机产生的一组模拟数据集进行分类。

解：

根据基于密度的聚类法的原理，编程计算，得到以下结果：

```
>> x=[randn(30,4)*0.4;randn(40,4)*0.5+ones(40,1)*[4 4 4 4]];
>> [class,type]=dbscan(x,2,[]);
```

从 class 值，可看出分类结果符合实际情况。从 type 值可看出，样本点都为核，不存在离群点，如图 22.7 所示（样本点的投影）。（因为数据是随机产生的，所以每次结果都不一定相同。）

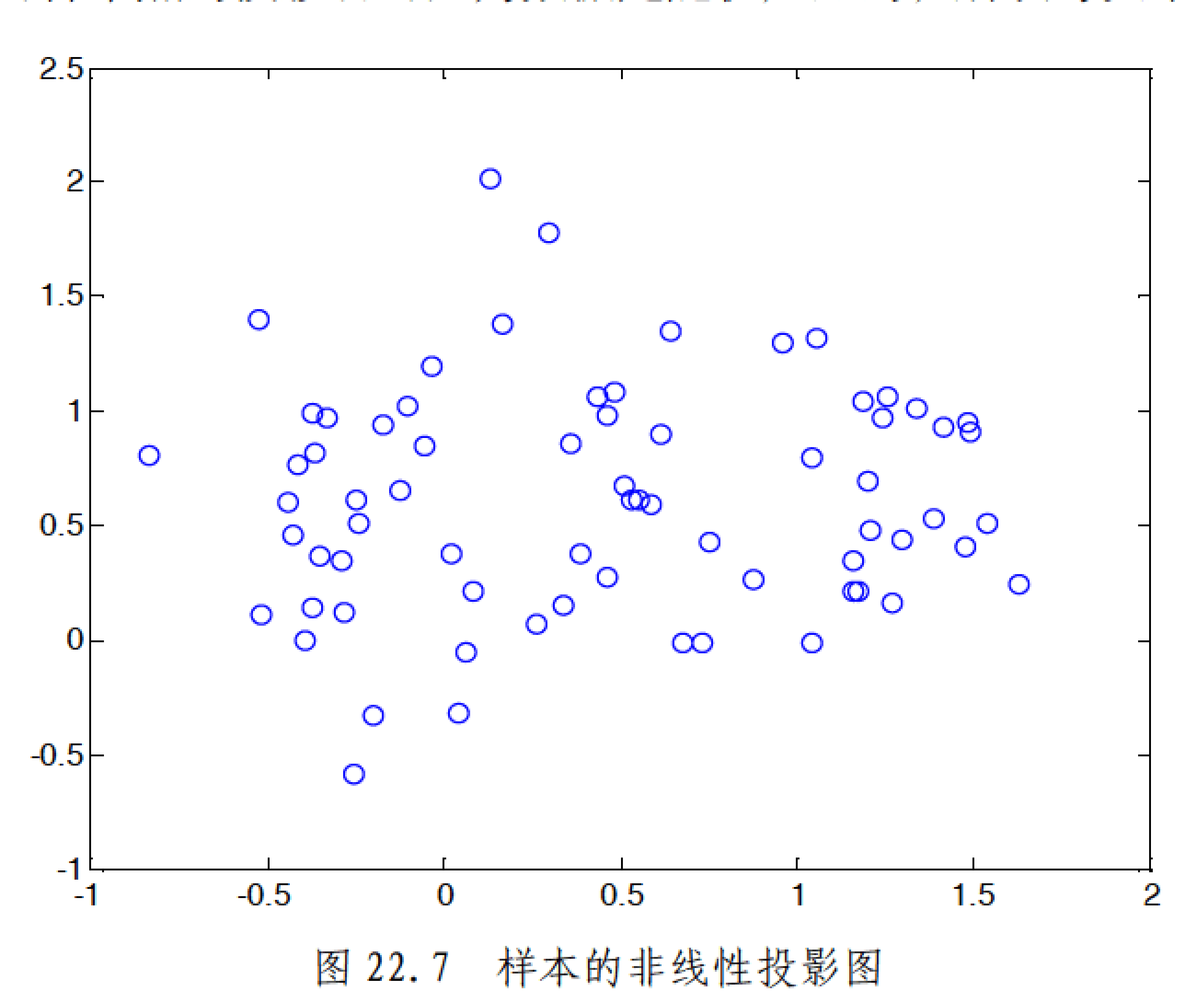


图 22.7　样本的非线性投影图

例 4.45　对例 4.39 中的表 22.11 的数据进行离群点分析。

解：

离群点的检测可以有多种方法，以下通过三种方法对此数据集进行分析：

（1）基于距离的离群点检测方法：

```
>>load data;
>>y=outlier(x,1);
y=3   13  14  12  2   %此 5 个点为最有可能的离群点
```

（2）基于相异度的离群点检测方法：

```
>> y=outlier_sim(x,[1 1 1 1])
  y=2  3   5  12  13    %此 5 个点为最有可能的离群点
```

此方法相异度的定义如下：

对于离散属性，当它们完全相等时，相异度为 0，否则为 1；

对于连续属性，相异度由下式定义：

$$\sum_{i=1}^{m}[(\frac{x1_i - x2_i}{x1_i + x2_i}) \times 2]^2$$

其中：m 为属性数目。

（3）基于聚类方法的离群点检测方法：

```
>> y=outlier_kmeans(x,3)            %基于 kmeans 聚类方法
 y=3                                %此点为最有可能的离群点
>> [class,p]=dbscan(x,3,[]);       %基于密度聚类方法
 p=outlier: [3 12 13 14]            %离群点
verge: [6 10 15]                    %边界点
```

从基于聚类的离群点检测方法结果可看出，此数据集的类别数有待商榷，3 点为真正的离群点。从图 22.8 也可以看出。

```
>>y=k_dist(data,1);                 %基于 k－近邻距离
>>y= k_dist(data,2);                %基于密度的方法
```

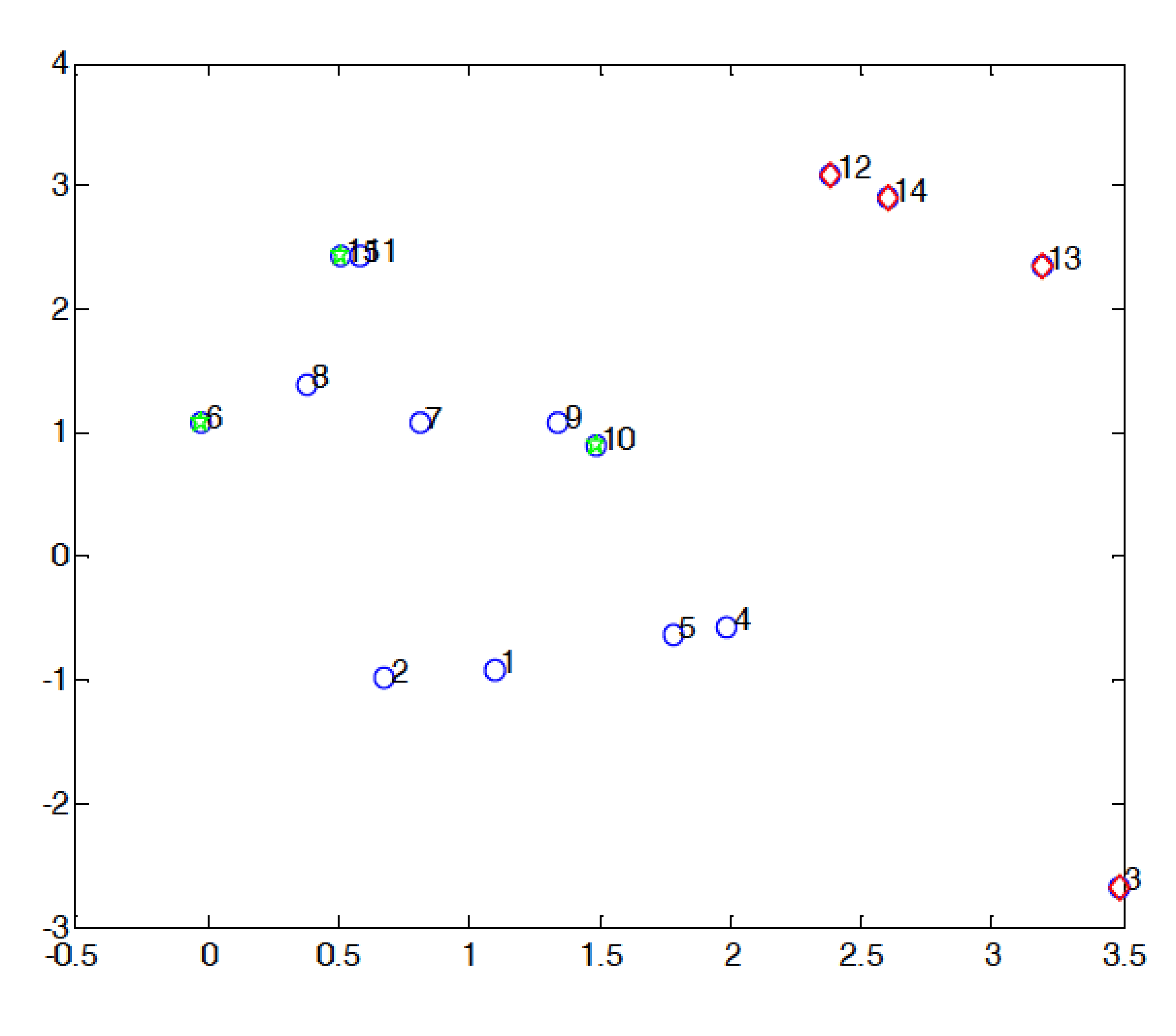


图 22.8　数据集的非线性投影图

这两种方法的结果都表明 3 号样品为异常点。

例 4.46　给定如表 22.11 所示的二维数据集，判断点 P15 为离群点的可能性。

表 22.11　二维数据集

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
x	1	2	2	2	3	4	5.5	5.5	6	6	6
y	7	8	7	6	7	7	6.5	7	8	7.5	7

	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22
x	6	6.5	6.5	7	7	7	2.5	3	3	4	5
y	6	7	6.5	8	7	6	2	1.5	2	5	4

解：

数据集的可视化图形如图 22.9 所示。

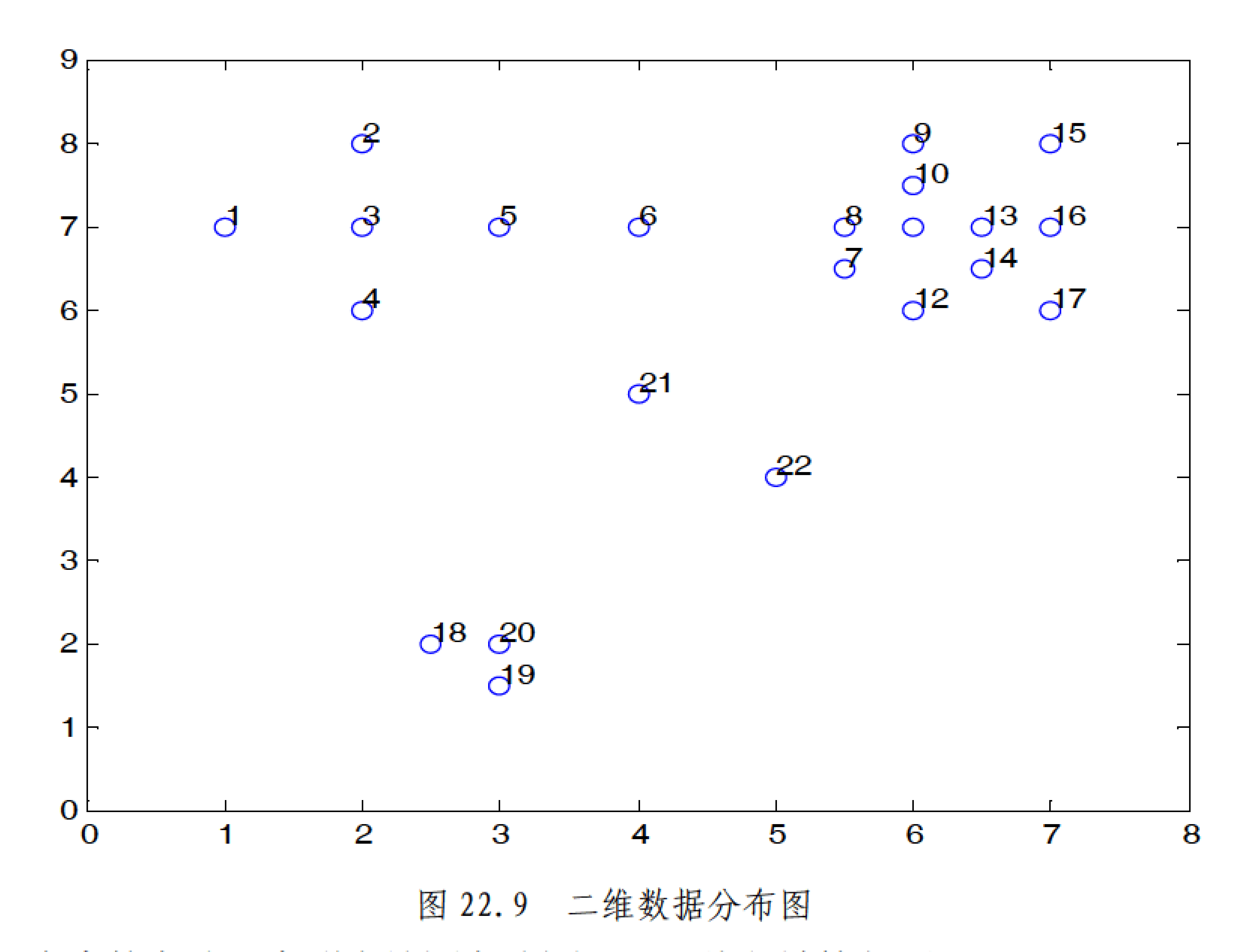


图 22.9　二维数据分布图

根据基于密度等方法的离群点检测方法原理，可编程计算如下。

```
>> data=[1 2 2 2 3 4 5.5 5.5 6 6 6 6 6.5 6.5 7 7 7 2.5 3 3 4 5;
         7 8 7 6 7 7 6.5 7 8 7.5 7 6 7 6.5 8 7 6 2 1.5 2 5 4]';
>> y=myoutlier(data,[0 0],2);
```

其中 y{1}为最近邻距离、y{2}为对象的相对密度，y{3}为离群度。从这 3 个数据看，点 P15 有可能为离群点，但其他点如 P22 等为离群点的可能性更大。

例 4.47　对于表 22.12 所示的二维数据集，比较点 P7 和 P11，哪个更有可能成为离群点。

表 22.12　二维数据集

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
x	1	1	1	2	2	2	6	2	3	5	5
y	2	3	1	1	2	3	8	4	2	7	2

解：

根据基于对象和簇的离群因子检测方法的原理，可编程计算如下。

```
>> data=[1 1 1 2 2 2 6 2 3 5 5;2 3 1 1 2 3 8 4 2 7 2 ]';type=[0 0];
>> [y3,y4,y5]=outlier_class(data,type);
```

其中 y3 为大于阈值的基于对象的离群点，y4 为基于对象的离群因子值，y5{1}为基于簇的离群因子，y5{2}为具有较大离群值簇中的数据点。

从结果及图 22.10 可看出，P7 点为离群点的可能性较大。

解：

数据集的可视化图形如图 22.9 所示。

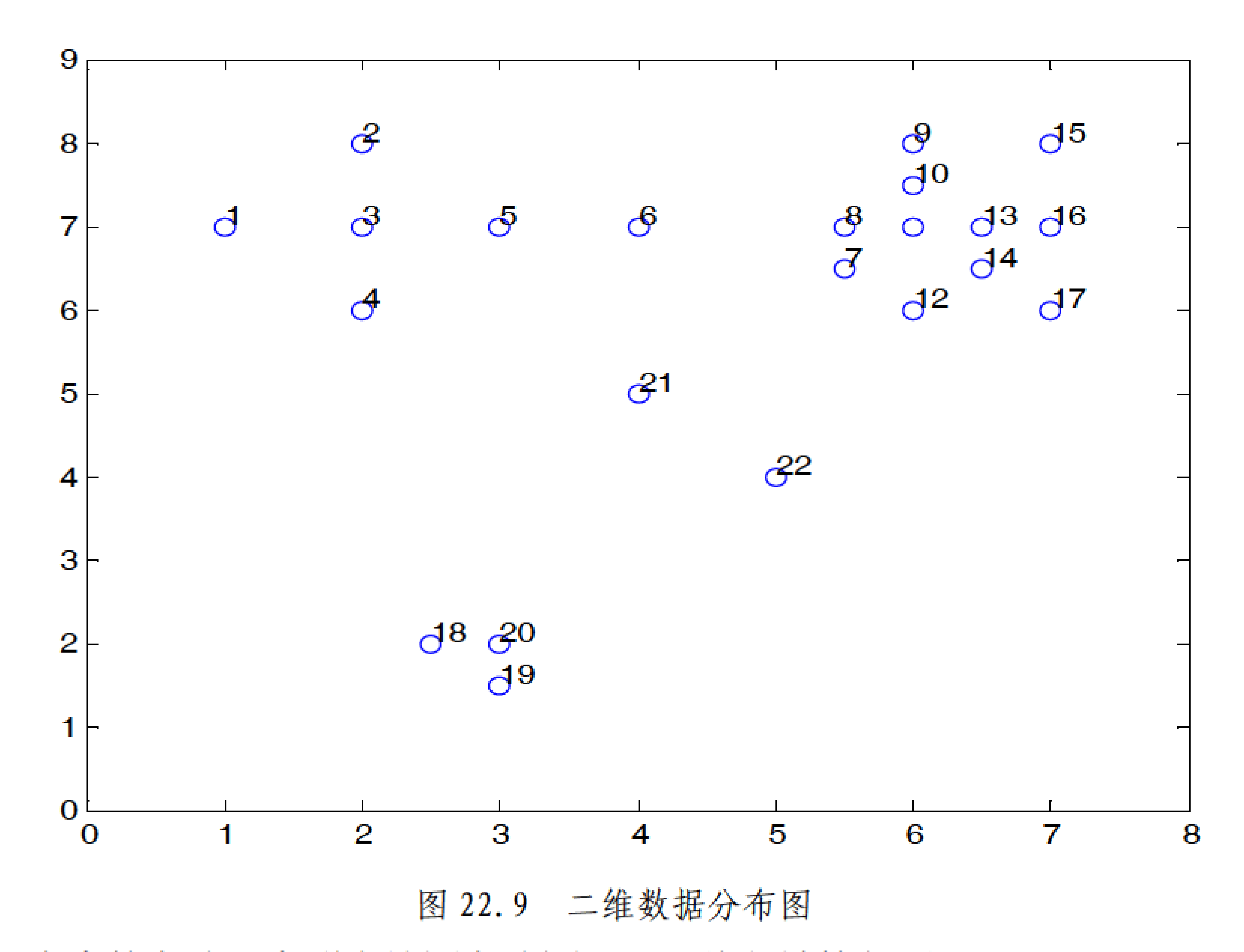


图 22.9　二维数据分布图

根据基于密度等方法的离群点检测方法原理，可编程计算如下。

```
>> data=[1 2 2 2 3 4 5.5 5.5 6 6 6 6 6.5 6.5 7 7 7 2.5 3 3 4 5;
         7 8 7 6 7 7 6.5 7 8 7.5 7 6 7 6.5 8 7 6 2 1.5 2 5 4]';
>> y=myoutlier(data,[0 0],2);
```

其中 y{1}为最近邻距离、y{2}为对象的相对密度，y{3}为离群度。从这 3 个数据看，点 P15 有可能为离群点，但其他点如 P22 等为离群点的可能性更大。

例 4.47　对于表 22.12 所示的二维数据集，比较点 P7 和 P11，哪个更有可能成为离群点。

表 22.12　二维数据集

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
x	1	1	1	2	2	2	6	2	3	5	5
y	2	3	1	1	2	3	8	4	2	7	2

解：

根据基于对象和簇的离群因子检测方法的原理，可编程计算如下。

```
>> data=[1 1 1 2 2 2 6 2 3 5 5;2 3 1 1 2 3 8 4 2 7 2 ]';type=[0 0];
>> [y3,y4,y5]=outlier_class(data,type);
```

其中 y3 为大于阈值的基于对象的离群点，y4 为基于对象的离群因子值，y5{1}为基于簇的离群因子，y5{2}为具有较大离群值簇中的数据点。

从结果及图 22.10 可看出，P7 点为离群点的可能性较大。

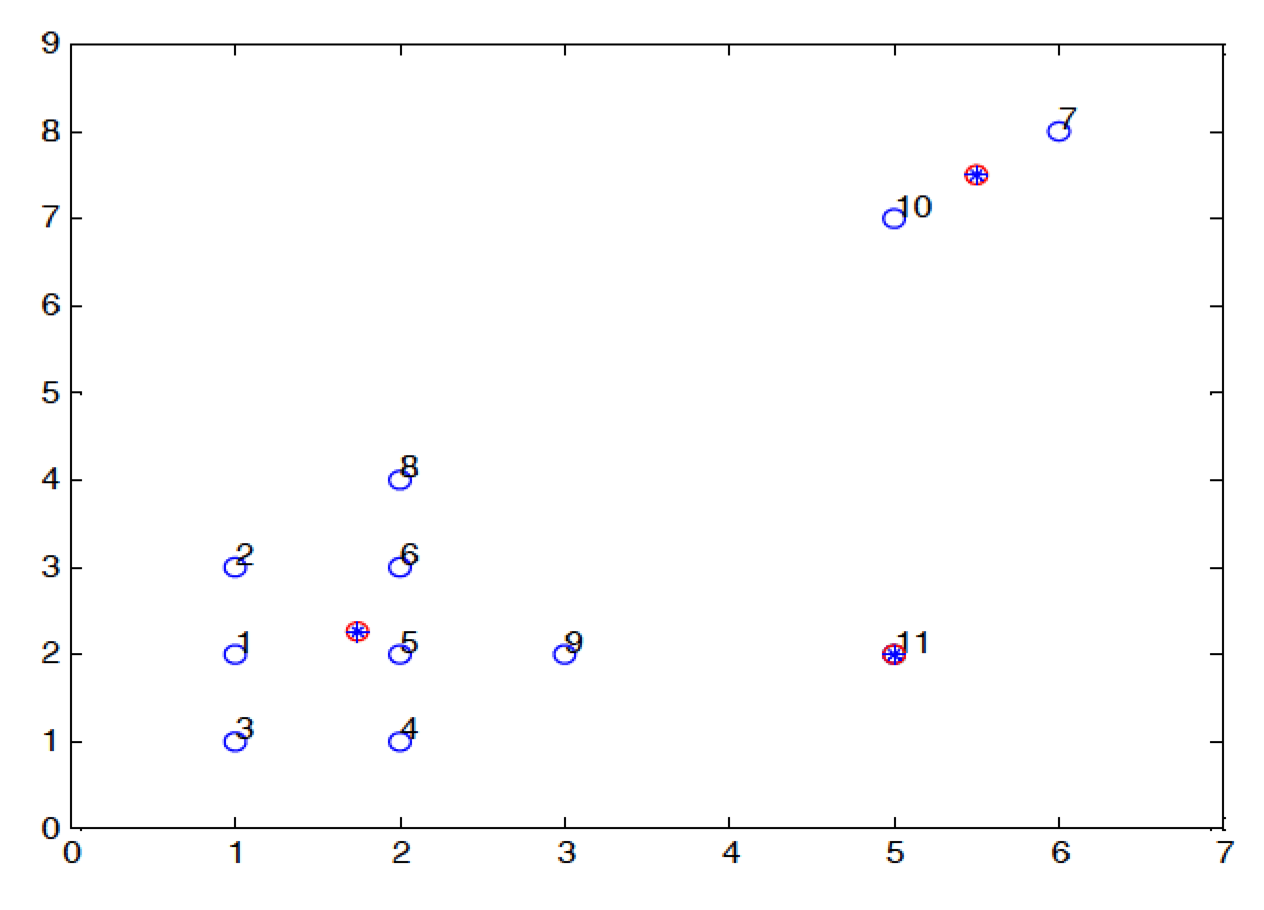


图 22.10　数据集及相应的类中心的分布图

例 4.48　在对数据集的聚类分析中，有些聚类方法对离群点以及初始取类中心比较敏感，如常用的 k-均值聚类方法。因此如能在这两方面进行改进，就会提高聚类结果。

请对 K-means 法进行改进，并对例 4.39 中的数据进行分析。

解：

根据以上两点，对 K-means 法进行修改，并编程计算如下。从结果及图 22.11 可看出，分类情况较为合理。

```
>>load data;
>>y=mykmeans(x,3);
 y.outlier: 3           %离群点
y.class:1  2  4  5  6  7  8  9  10  11  12  13  14  15
        1  1  1  1  2  2  2  2   2   2   3   3   3   2
```

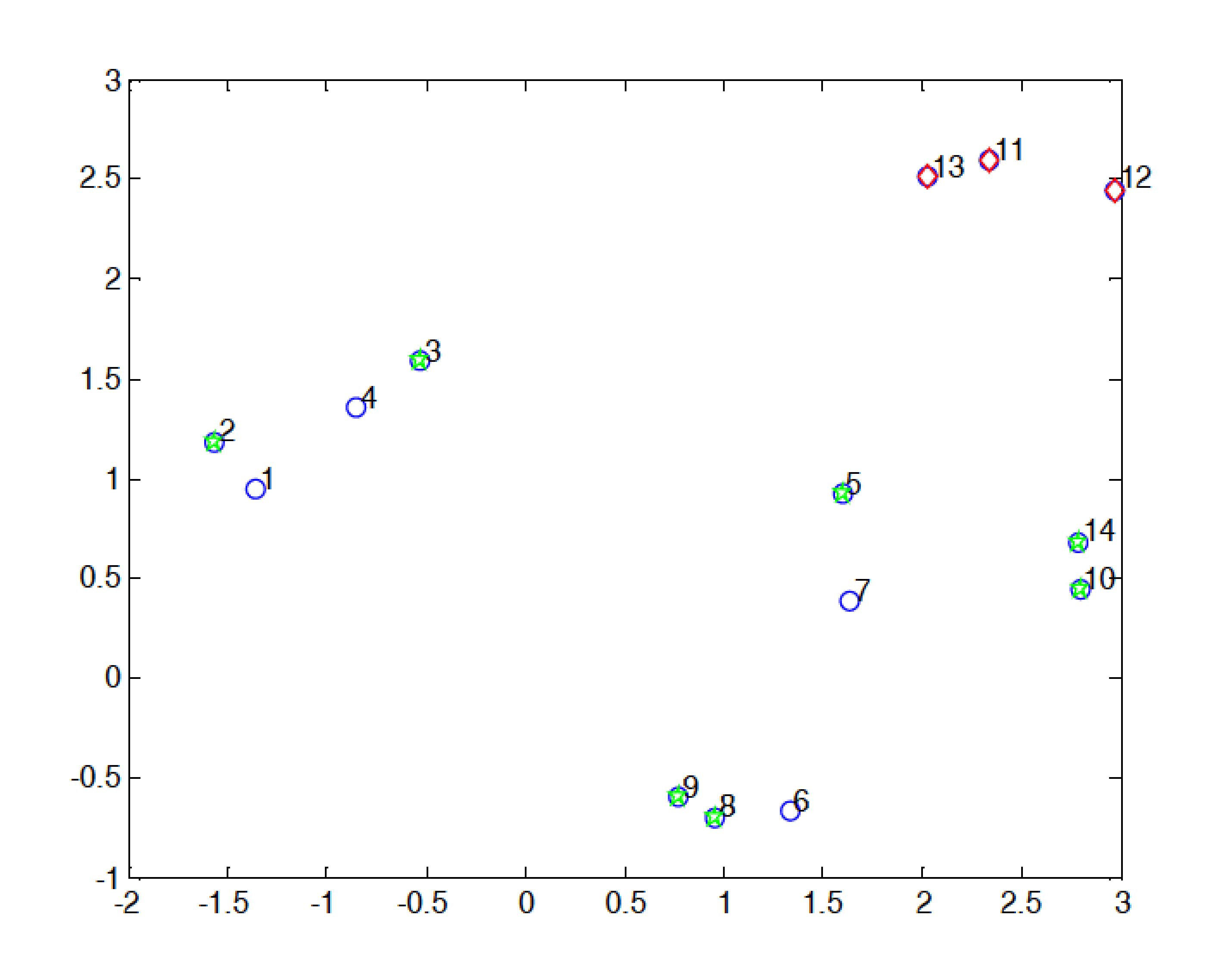


图 22.11　数据集的非线性投影图

例 4.49　在聚类分析中，聚类数直接影响到聚类效果。在实际中，可能没有办法得到确切的聚类数，或指定的聚类数不一定是最佳的聚类数。为此，需要用一定的方法来确定较佳的聚类数。

请对例 4.39 中的数据确定最佳聚类数。

解：

可以用试探法等多种方法确定聚类数。下面设定以下的指标，通过计算在不同聚类数情况下的聚类结果（指标值），以确定较佳的聚类数。

指标由两部分组成，一是类内距离 b，其定义为类内某一样本到其他样本间的距离和的平均值

$$b(j,i)=\frac{1}{n_j}\sum_{q=1}^{n_j}\|X_q^j-x_i\|^2$$

式中：x_i 为样本集 j 类中的某一样本；X_q^j 为该类中的其他样本。

二是类间距离，它为某一样本到其他类样本的距离和平均值的最小值，所以称为最小类间距离

$$w(j,i)=\min(\frac{1}{n_k}\sum_{q=1}^{n_k}\|X_q^k-x_i^j\|^2)$$

式中：x_i^j 为 j 类的某一样本；X_q^k 为 k（$k\neq j$）类中的样本。

则此指标（BWP）为下式

$$\mathrm{BWP}(j,i)=\frac{\mathrm{BSW}}{\mathrm{BAW}}=\frac{b(j,i)-w(j,i)}{b(j,i)+w(j,i)}$$

求出数据集每个数据的 BWP 指标值，并将它们的平均值作为聚类性能指标。此值越大，聚类效果越好。

根据以上指标的定义，可编程计算如下，计算过程中所用的最大聚类数，可以指定或为 $\mathrm{int}(\sqrt{n})$，n 为数据集中的样本数。

```
>>load data;
>> y=clustingk(data)  %即分为 4 类，效果可能更好
   y= 4
```

例 4.50　对表 22.13 中的数据，利用一趟聚类算法对其进行聚类。

表 22.13　天气气象情况

outlook	temperature	humidity	windy
sunny	85	85	false
sunny	80	90	true
overcast	83	86	false
rainy	70	96	false

续表

outlook	temperature	humidity	windy
rainy	68	80	false
rainy	65	70	true
overcast	64	65	true
sunny	72	95	false
sunny	69	70	false
rainy	75	80	false
sunny	75	70	true
overcast	72	90	true
overcast	81	75	false
rainy	71	91	true

解：

根据一趟算法的原理，可编程计算如下。

```
>> x={'sunny' 85 85 'false';'sunny' 80 90 'true';'overcast' 83 86 'false';
      'rainy' 70  96  'false';'rainy' 68  80  'false';'rainy' 65 70 'true';
      'overcast' 64 65 'true';'sunny' 72 95 'false';'sunny' 69 70 'false';
      'rainy' 75 80 'false';'sunny' 75 70 'true';'overcast' 72 90 'true';
      'overcast' 81 75 'false';'rainy' 71 91 'true'};type=[1 0 0 1];
>> class=ridecluster(x,type,[]);
>> class{1}=1  2  3  13      %class 为分类结果
>> class{2}=4  5  8  10  12  14
>> class{3}=6  7  9   11
```

一趟聚类算法具有近似时间复杂度，其本质类似于 K-means 算法，不能用于发现非凸形状的簇，或具有各种不同大小的簇。对于具有任意形状的簇的数据集，算法可能将一个大的自然簇划分成几个小的簇，而难以得到理想的聚类结果。

从计算结果分析，此算法对样本的顺序、聚类阈值及样本是否规范化等比较敏感。对于大规模数据集的聚类可以采用类似 BIRCH 算法的两阶段聚类思想，结合一趟算法的高效性及其他可识别任意形状簇的聚类算法的优点得到混合聚类算法。如选取较小的阈值，利用一趟算法产生初始聚类，将得到的簇作为整体看成对象，再利用其他算法进行聚类，可以得到很好的结果。

第 23 章

时序数据挖掘

23.1 基本定义

时序数据是指数据库中保存的是大量时间点上的数据。例如卫星收集的图像和传感器数据、病人的心电图等。时序数据经常涉及时间段，即一个开始时间和一个结束时间。每个记录都与一个区间$[t_s,t_e]$相关联，这里 t_s 是开始时间，t_e 是结束时间。这个区间通常称为有效时间。另一个可能用到的时间是事务时间，它是指插入记录的事务所关联的时间戳，它与有效时间的起始点可能不同。例如某人在 2009 年 1 月 8 日表明他将在 2009 年 3 月 9 日有一个新住址，那么新住址的有效时间的起始点是 2009 年 3 月 9 日，但是事务时间是 2009 年 1 月 8 日。

从时间的角度考虑，至少存在 4 种类型的时序数据库：

① 快照数据库：数据库中存储的数据是在当前时间内有效的数据。

② 事务时间数据库：数据库支持的时序数据只包括与插入数据事务相关联的时间，它可以是事务被提交（或被请求）时的一个时间戳，也可能是一个时间范围。

③ 有效时间数据库：数据库支持的数据的有效时间范围。有效时间可以用单值表示，也可以用一个区间表示。如果用单值，这个值就是有效时间范围的起始点，它的结束点是具有键值的数据的下一个时间范围的起始点。

④ 二重时间数据库：数据库既支持事务时间又支持有效时间。

时序数据挖掘包括许多常规的数据挖掘方法，但是，它自然也因时序方面的复杂性和更复杂的查询类型而变得更为复杂。

时序序列是项集的有序排列。给定一个顾客事务（交易）的数据库 D，每个事务由下列字段组成：顾客标记、事务时间及在事务中所购买的商品项。在同一时间不存在一个顾客多于两个以上的事务发生，在事务中不考虑所购买项目的数量，即只关心一个项目被购买还是没有被购买。一个顾客的所有事务放在一起可看作是一条序列。

时序序列和时间序列的一般区别是时序序列不必与时间有明显的关系，它只要求序列中的项必须是完全有序的。事实上，时序序列和时间序列这两个词是可以互相替换的。这两个概念的基本区别是时间序列是数值的有序排列，而时序序列则是项或数值的集合的有序排列。时序序列的长度是序列中所有项集的基数和。一个给定时序序列的子序列是从原序列中移去一些项，并移去由此产生的所有空项集而得到的序列。

一个项集 i 可以定义为（$i_1i_2\cdots i_m$），其中 i_j 是一个项，一个序列如 s 为$<s_1s_2\cdots s_n>$，其中 s_j 是一个项集。

给定两条序列 $A=<a_1a_2\cdots a_n>$和 $B=<b_1b_2\cdots b_m>$，$m\leqslant n$，若存在一组整数 $i_1<i_2<\cdots<i_n$，且 $a_1\subseteq b_1, a_2\subseteq b_2,\cdots,a_n\subseteq b_m$，则称序列 B 包含序列 A，或称 A 是 B 的子序列。例如，序列<(3)(4 5)(8)>∠<(7)(3 8)(9)(4 5 6)(8)>，其中符号“∠”代表“被包含于”的关系。因为 $(3)\subseteq(3\ \ 8),(4\ \ 5)\subseteq(4\ 5\ 6),(8)\subseteq(8)$，但是，序列<(3)(5)>并不被包含于<(3 5)（反之也真），前者代表项 3 及 5 是在购买一段时间之后再购买另一个，而后者代表两个项同时购买。在一条序列集合中，若一条序列不包含在任何其他序列中，则称其为最大序列。

时序数据挖掘就是在所有满足由用户指定的最小支持度阈值的序列中发现最大序列，每个这种最大序列代表一个序列模式。称满足最小支持度限制的一个序列是一个频繁序列。随着序列模式挖掘的应用越来越复杂，为了更有效地挖掘出用户感兴趣的序列模式，而不是仅仅局限

在挖掘出所有满足支持度的频繁序列上，可以更需要挖掘出用户感兴趣的特定模式。用户的兴趣往往会被转化成一种约束，用来限定序列模式，即只有满足这种约束的序列模式才是用户真正需要的模式。

经过多年研究，目前已有多种挖掘时序数据的方法，归结起来，基本可以分为两大类：通过生成候选并利用 Apriori 性质进行剪枝的 Apriori 类算法，以及通过时序数据库进行划分的模式生成算法。

23.2　时序数据挖掘参数

大多时序数据挖掘研究都集中在符号模式方面，这是因为数值曲线模式通常是属于趋势分析和时序数据的统计预测范畴。

时序数据挖掘涉及一些参数的设置，这些对数设置的好坏对时序数据挖掘结果影响很大。不同时序数据挖掘任务需要设置不同的参数。

1. 时间长度

可以将数据库中的整个时序或用户所选择的序列作为时间序列的长度。时序数据挖掘将仅限于在这一序列长度之内进行。此外，也可以将序列长度看成是由若干序列的成分，如，每年、每月或火山爆发前每两周等，这种情况下，就能够发现周期性序列模式。

2. 事件窗口

一系列在一段时间内发生的事件在特定的分析中可以看成是一起发生的。如果一个事件窗口 D 被设置为同序列 H 一样长，那就会发现对时间不敏感的频繁模式，也就是基本模式。如“2012 年，一个购买计算机的客户也买了数码相机”（其中不再关心先买哪个后买哪个）；若一个时间窗口 D 被设置为 0，那就会发现一个序列事件是作为单个事件发生。例如“一个顾客购买了计算机，然后又购买了内存，最后会购买 CD－ROM”。若一个事件窗口 D 被设置为上述两者之间的某个值（即 0 与总长度之间），如若 D 设为一个月，那么在同一个月发生的交易事务，将被认为是在同一时间发生的，而被合在一起进行分析。

3. 时间间隔

如果时间间隔 interval 设为 0，就意味着没有间隔，也就是发现严格的连续时间序列。发现序列模式 $a_{i-1}a_ia_{i+1}$，这里 a_i 就是在时间 i 发生的事件。这里也可以将事件窗口考虑进来，如若事件窗口设为一周，那么也就是要发现连续各周频繁（发生）模式。

23.3　时序关联规则

在传统的关联规则中，一个事务可以看作是具有以下形式

$$<TID,CID,I_1,I_2,\cdots,I_m>$$

其中：TID 是事务的标识，CID 是客户的标识，$I_1,I_2,\cdots,I_m$ 称为项。当考虑时序数据数据库时，一个事务可以记为

$$<TID,CID,I_1,I_2,\cdots,I_m,t_s,t_e>$$

其中：$[t_s,t_e]$表示事务的有效时间范围。例如对于一个商店所产生的事务，可以设定 $t_s=t_e$ 为事务完成的时间。对于互联网上的订货事务，可以分别将 t_s 和 t_e 设定为订货时间和实际的交货时间。随着数据库加入新的时间数据，就可以发现处在不同时间点或时间段上的不同关联规则。

23.3.1 事务间关联规则

基本的关联规则只关注一个事务内的项之间的关联规则，这类规则称为事务内关联规则。但还有一些跨事务的规则也很有意义。例如一个电子商店的经营者想知道顾客在购买了计算机之后是否会购买计算机软件。这种购买活动将发生在两个不同时间的事务中，即存在新规则—事务间关联规则。为了定义这些新规则，可以将窗口的概念应用于事务数据库，即在基本的关联规则问题中存在一个项集 $I=（I_1I_2\cdots I_m）$和一个事务数据库 $D=（t_1t_2\cdots t_n）$，其中 $t_i=I_{i1},I_{i2},\cdots,I_{ik}$，$I_{ij}\in$I，设每个事务 t_i 与一个值 d_i 相关联，这个值可以是时间、位置或其他描述事务的信息。如果是时序数据，则这个值是时间。

23.3.2 情节规则

情节规则是应用于事件序列的扩展的关联规则。事件序列 S 是一个有序事件列表，其中每个事件都发生在一个特定的时间。因此，它是特殊的时间序列。情节（episode）是一系列事件谓词 A 和 A 中事件的偏序≤的集合：{A，≤}。当事件谓词应用于一个实际事件时，它要么被评价为真，要么被评价为假。情节可以看作是一个有向图，其中顶点表示事件，弧表示偏序。如果情节 B 的有向图是情节 A 的有向图的子图，则情节 B 称为情节 A 的子情节。如果事件序列 S 中的所有告警谓词都得到满足，并且这些事件符合偏序规则，那么事件序列 S 包含一个情节。情节规则的形式化定义如下：情节规则具有 B ⇒ A 的形式，其中 B 和 A 是情节，且 B 是 A 的子情节。

情节规则可以用来预测交换机节点的故障。当用时序数据挖掘的角度来看这个问题时，它就转化成基于前面的事件序列来预测一个事件（故障）的问题。这个事件可以看作是通过一个结点的流量或警报（由网络实体生成的描述一个问题的消息）。警报可以看作是一个元组 $\alpha=<t,s,m>$，其中 t 是警报发生的时间，s 是警报的来源，m 是警报消息的本身。警报序列可以看作是一个时间序列。首先通过一些预处理技术实现去除冗余警报、去除比已存在的警报优先级低的警报，用新信息或更高级的警报替换某些警报等操作，然后用相关模式来匹配在警报数据中发现的序列。将这个模式与最近时间窗口内发生的警报进行比较，如果产生的警报序列与一个相关模式匹配，那么与它相关联的相关行为就将发生，即预测故障点。

23.3.3 序列关联规则

给定一个项的集合 $I=（I_1I_2\cdots I_m）$和由客户序列中的按客户分组的一组事务的集合，序列关联规则具有 $S\Rightarrow T$ 的形式，其中 S 和 T 是序列，其支持度是包含序列和的客户（客户序列）所占的百分比。置信度是包含序列 S 和 T 的客户（客户序列）的个数与包含序列 S 的客户（客户序列）的个数之比。

根据以上定义，就可将序列关联规则问题描述为：已知最小支持度和置信度，找出序列关

联规则。实际生活中有很多应用会用到序列关联规则。例如在购物篮分析领域，随时间变化的购买行为可以用来预测将来的购买行为，进而可以利用预测出的最可能购买行为来制作针对客户的广告。

23.3.4　日历关联规则

给定一个项的集合 $I=(I_1I_2\cdots{}_m)$，一组事务的集合 $D=(t_1t_2\cdots t_n)$，一个时间段 k 和一个日历 $C=\{(s_1,e_1),\cdots,(s_k,e_k)\}$，日历关联规则是发生在 $D[k]$中的关联规则 $X\Rightarrow Y$。在该问题中每个事务 t_i 执行时都与一个时间戳 t_{is} 相关联，另外，整个时间按照预先定义的时间单位 t 进行划分。时间段 k 定义为区间$[kt,(k+1)t]$；如果事务 t_i 的时间戳满足 $kt\leqslant t_{is}\leqslant(k+1)t$，则称该事务在时间段 k 内发生。$D[k]$是发生在时间段 k 内的事务的集合，$D[k]$中的项集 X 的支持度是 $D[k]$中包含 X 的事务所占的百分比，$D[k]$中 $X\Rightarrow Y$ 的置信度是 $D[k]$中包含 $X\cup Y$ 的事务的个数与包含 X 的事务的个数之比。要注意的是，相同的数据可以按不同粒度的时间单位（如小时、月、年等）来挖掘日历关联规则。

23.4　时间序列挖掘

时间序列是在一段时间上的一组属性值。时间序列数据可以是连续的，也可是离散的。

针对时间序列的数据挖掘应用包括：度量两个不同时间序列的相似性；给定一个具有一组已知值的时间序列，预测属性的未来值。显然第二种应用的预测是一种分类，而第一种确定相似性可以认为是聚类或分类。

23.4.1　时间序列分析

时间序列分析可以认为是从数据中发现模式并预测未来值的过程，其中模式识别可能涉及：

- 趋势　趋势可以看作是属性值随时间进行的、系统的、无重复的改变（线性或非线性）。
- 周期　指时间序列中的行为具有周期性。
- 季节性　检测的模式可以是基于年、月、日这样的时间点。
- 异常点　为方便模式识别，需要技术来剔除或减少异常点的影响。

为方便模式识别，实际的时间序列数据可以要作某种形式的变换，如对数变换可以用来稳定变化并使季节性作用恒定。另外，变换也可以用来解决维数灾难问题，带有很多变量的时间序列数据挖掘不仅困难而且代价高昂，存储高维数据的数据结构也常常不够高效，此时就需要进行变换以达到降维的目的。

23.4.2　趋势分析

时序变量 Y 表示时间的函数，即 $Y=F(t)$，此函数可以图示为一个时序图，如图 23.1 所示，它描述了某地区每月的降水量。

目前一般有四种主要的变化或成分用于刻画时序数据：

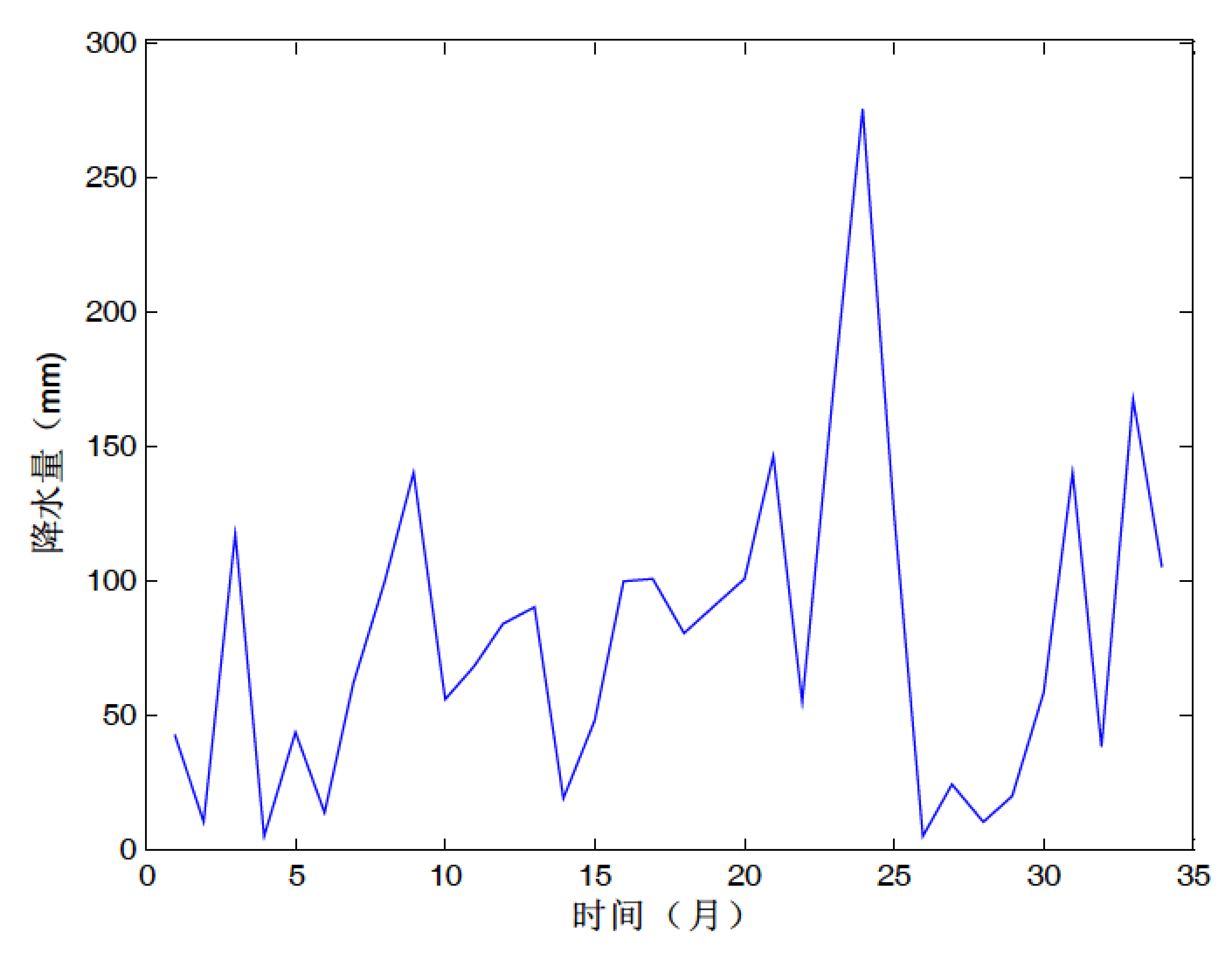


图 23.1　某地区降水量

1. 长期或趋势变化

趋势可以看作是属性值随时间进行的、系统的、无重复的改变（线性的或非线性的），主要用于反映时间序列图在长时间间隔运动的一般变化方向。这种变化反映为一种趋势线或趋势曲线。确定趋势曲线或趋势线的典型方法包括加权移动平均法和最小二乘法。其中，移动平均法对一个围绕特定时间点的时间窗口，用窗口内的所有取值的局部平均值来代替这一点的真实值。

2. 周期运动或变化

周期主要指循环型，即趋势线或曲线在长期时间内呈摆动迹象，它可以是也可以不是周期性的。即在等时间间隔之间，周期不需要严格遵循类似模型演进。

3. 季节性变化或变化

反映每年都重复出现的事件，如电煤的库存在夏、冬季会非常紧张。也就是说，季节性变动是指同一或近似同一的模式，在相继年份的对应月份或时期重复出现。

4. 非规则或随机变化

反映由于随机或偶然事件引起的零星时序变化，如地震、企业人事变化等。

以上有关趋势、周期、季节性和非规则的变动，可以分别用变量 T、C、S 和 I 表示。时序分析与建模也可以将时序分解为以上 4 个基本运动。时序变量 Y 可以为 4 个变量的乘积或求积。通过对趋势、周期、季节和非规则成分的变动的系统分析，人们可以合理预测长期或短期信息。

可以用很多直观的方法检测时间序列中存在的趋势。平滑就是一种去除时间序列中非系统化行为的方法。平滑常常是通过发现属性值的移动平均来实现的，即给定一个围绕特定时间点的时间窗口，移动平均用窗口内所有取值的局部平均值来代替这一点的真实值。这里，经常用到的是

中位数而不是均值。平滑也可以用来预测未来值，因为平滑后的数据较容易满足一个已知的函数，如线性函数、对数函数、指数函数等。

识别时间序列数据中的季节性模式相对困难。一种方法是在均匀分隔的时间段上检测属性之间的相关性。相关项之间的时间差称为时滞。自相关函数用来确定不同时滞间隔上的数据的相关关系。相关图直观地给出了不同时滞值的自相关值。

协方差用来度量两个变量的同步变化程度，可以以它为依据确定两个时间序列之间或一个时间序列中的季节性趋势之间的关系。自相关系数用来度量时滞间隔为 k 的时间序列值之间的相关性。

23.4.3　相似性搜索

相似性搜索用于找出数据库中与给定查询序列最相近的数据序列。给定时间序列集合 S，主要可以进行子序列匹配和全序列匹配两种类似的相似性，其中子序列匹配是找出与给定的查询序列相似的所有数据子序列，而全序列匹配是找出彼此间相似的序列集合，子序列匹配是应用中更常遇到的问题，在金融市场分析、医疗诊断分析和科学与工程数据库分析等，时序分析中的相似性搜索具有非常广泛的应用。

然而，由于时间序列数据规模大、维度高、自身带有时间性和数值波动大的特点，为数据挖掘带来困难，通常，在数据挖掘前需要进行数据规约和变换，从而缩小存储空间，加快处理速度。

1. 数据变换

许多信号分析处理技术要求数据在频率域中，以便应用欧式距离等各种度量方式，因此常常需要进行数据变换，将时间序列从时间域变换到频域，常用的是正交交换。另外还可以使用独立于数据的变换，如离散傅里叶和离散小波变换，通常这些变换可以有效地解决高维特征向量的降维问题。

2. 相似搜索的索引方法

时间序列数据经过适当变换后，为提高访问效率，可以用头几个傅里叶系数构建一个多维索引。当相似查询提交给系统后，可以利用索引检索出与查询序列保持一定最小距离的序列。通过计算时间域序列和未满足查询的序列间的实际距离，可以进行必要的后处理。

为了在大型数据库中改进相似性搜索的效率，通常将索引建成树形结构。在树的高层，划分比较粗略；在树的低层，划分较为细致。

在进行序列匹配时，首先被每一序列分割为长度为 w 的窗口“片段”，然后，将每个序列映射为特征空间中的一个“轨迹”，接着，将每个序列的轨迹划分为“子轨迹”，每一个由最小边界矩形表示。利用多片组装算法可以搜索更长的匹配序列。

3. 处理偏移和振幅中间隙和差异的相似性搜索方法

序列相关性分析通常使用欧氏距离作为相似性度量，距离越小，两个序列越相似，但在实际应用中，不一定要求匹配的子序列在时间轴上完全一致，亦即若子序列对具有同样形状，但在序列内存在间隙（其中一个序列的某些值可能在另一个序列中缺失）或在偏移或振幅中差异，也认为它们是匹配的，这为在包括股票市场和心脑电图分析中的应用提供了工具。

当两序列之间存在足够多的非重叠的、相似的、时间有序的子序列时，可以判断子序列相互匹配。据此，处理偏移与振幅中间隙和差异的相似搜索可以如下执行：①原子匹配，规范化数据，寻找所有无间隙的较小相同窗口对；②窗口匹配，将相同窗口缝合，形成大的相似子序列对，其中允许在原子匹配间有间隙；③子序列排列：线性排列子序列匹配，以判定是否存在足够多的相似片段。通过这些处理，能够发现相互匹配或与查询模板匹配的形状相似，但是具有间隙或偏移时和振幅存在差别的序列。

23.4.4 周期分析

周期分析是对周期模式的挖掘，即在时序数据库找出重复出现的模式。周期模式可以应用诸如行星运动规律分析、潮汐、每日能耗等许多重要的领域。

周期模式挖掘的问题可以分为三类：①全周期模式，每一时间点都精确或近似影响着时序上的循环行为。②部分周期模式，它描述在部分时间点的时序周期，部分周期是一种比全周期较为松散的形式，在现实世界中更为常用。③循环或周期相关联规则，这种规则是周期出现的事件的关联规则。

23.5 时间序列分段线性表示

作为时间序列描述的一种方法，分段线性化表示通过对时间序列的简化及近似表示来压缩原始时间序列，换来更小的存储和计算代价；保留时间序列的主要形态，去除细节干扰，有利于提高挖掘的效果和准确性等。

假设时间序列 T 有 n 个数据点，首先将其分成 k 个直线段，k=3/n，当 n 不是 3 的整数倍时，允许最后一段包含 4 个或 5 个点。每个直线段与原始数据之间都有一个残差，这个残差用直线段和原始数据点之间的垂直距离表示，定义为 $d_1,d_2,\cdots,d_i$，用规范化形式表示第 m 段的误差 e_m，有

$$e_m = \frac{\sum_{l=1}^{i} d_l^2}{i}$$

e_m 表征了直线段与原始数据之间的近似程度，对于由 k 个直线段近似的时间序列，定义其误差为 B_k=std($e_1,e_2,\cdots,e_k$)(std 为标准差)。然后合并两个近邻的直线段，直到用一个直线段来近似时间序列，即 k=1,2,$\cdots$,n/3，则此时需要考虑两个问题：①如何选择需要合并的邻近直线段；②哪个近似是最终需要的。

两个直线段合并的原则是使合并后的 B_k 最小，同时选择 B_k 最小的近似作为最终的近似。

分段线性化之后，可以对每一个直线段赋予一个权值来表明该直线段对整个时间序列波形所起的作用。一个时间序列若用 K 个直线段描述，其直线段序列用 A 表示，则 A 将是一个长度为 K 的 5 元向量 $A = \{AXL, AXR, AYL, AYR, AW\}$。

A 中第 i 个直线段由左端点(AXL_i,AYL_i)和右端点(AXR_i,AYR_i)以及权值 AW_i 表示。

基于此分段线性化表示方法，可以用下式测量时间序列的相似性，此公式即为两个直线段序列之间的距离。

$$D(A,B) = \sum_{i=1}^{K} AW_i * BW_i * (|AYL_i - BYL_i| - |AYR_i - BYR_i|)$$

当两序列之间存在足够多的非重叠的、相似的、时间有序的子序列时，可以判断子序列相互匹配。据此，处理偏移与振幅中间隙和差异的相似搜索可以如下执行：①原子匹配，规范化数据，寻找所有无间隙的较小相同窗口对；②窗口匹配，将相同窗口缝合，形成大的相似子序列对，其中允许在原子匹配间有间隙；③子序列排列：线性排列子序列匹配，以判定是否存在足够多的相似片段。通过这些处理，能够发现相互匹配或与查询模板匹配的形状相似，但是具有间隙或偏移时和振幅存在差别的序列。

23.4.4 周期分析

周期分析是对周期模式的挖掘，即在时序数据库找出重复出现的模式。周期模式可以应用诸如行星运动规律分析、潮汐、每日能耗等许多重要的领域。

周期模式挖掘的问题可以分为三类：①全周期模式，每一时间点都精确或近似影响着时序上的循环行为。②部分周期模式，它描述在部分时间点的时序周期，部分周期是一种比全周期较为松散的形式，在现实世界中更为常用。③循环或周期相关联规则，这种规则是周期出现的事件的关联规则。

23.5 时间序列分段线性表示

作为时间序列描述的一种方法，分段线性化表示通过对时间序列的简化及近似表示来压缩原始时间序列，换来更小的存储和计算代价；保留时间序列的主要形态，去除细节干扰，有利于提高挖掘的效果和准确性等。

假设时间序列 T 有 n 个数据点，首先将其分成 k 个直线段，k=3/n，当 n 不是 3 的整数倍时，允许最后一段包含 4 个或 5 个点。每个直线段与原始数据之间都有一个残差，这个残差用直线段和原始数据点之间的垂直距离表示，定义为 $d_1,d_2,\cdots,d_i$，用规范化形式表示第 m 段的误差 e_m，有

$$e_m = \frac{\sum_{l=1}^{i} d_l^2}{i}$$

e_m 表征了直线段与原始数据之间的近似程度，对于由 k 个直线段近似的时间序列，定义其误差为 B_k=std($e_1,e_2,\cdots,e_k$)(std 为标准差)。然后合并两个近邻的直线段，直到用一个直线段来近似时间序列，即 k=1,2,$\cdots$,n/3，则此时需要考虑两个问题：①如何选择需要合并的邻近直线段；②哪个近似是最终需要的。

两个直线段合并的原则是使合并后的 B_k 最小，同时选择 B_k 最小的近似作为最终的近似。

分段线性化之后，可以对每一个直线段赋予一个权值来表明该直线段对整个时间序列波形所起的作用。一个时间序列若用 K 个直线段描述，其直线段序列用 A 表示，则 A 将是一个长度为 K 的 5 元向量 $A = \{AXL,AXR,AYL,AYR,AW\}$。

A 中第 i 个直线段由左端点(AXL_i,AYL_i)和右端点(AXR_i,AYR_i)以及权值 AW_i 表示。

基于此分段线性化表示方法，可以用下式测量时间序列的相似性，此公式即为两个直线段序列之间的距离。

$$D(A,B) = \sum_{i=1}^{K} AW_i * BW_i * (|AYL_i - BYL_i| - |AYR_i - BYR_i|)$$

23.6　时间序列的预测

对于时间序列的预测可以用前面描述过的技术，如回归。但是在实际应用中时间序列数据往往存在着误差和噪声，使用简单的回归并不能达到满意的结果。

研究时间序列预测时常常假设时间序列是平衡的，这就意味着序列中的值来自一个均值不变的模型。更复杂的预测技术可以假设时间序列是非平稳的。

时间序列通常代表一些互相依赖的值，但它们可以看作是由一连串称为冲击的独立的值构成。这些冲击随机取自一个均值为 0 的正态分布。可以认为这些随机值构成的序列代表了一个白噪声过程。

这个白噪声过程可以通过一个线性过滤器转变成一个时间序列，其中线性滤波器可以看作是对前面冲击的生产关系加权和。

一个特殊的线性滤波器模式假设时间序列中的值依赖于它前面的元素的值。那么就可以用自回归技术。通过前面的元素值来预测时间序列的未来值，即

$$x_{n+1} = \xi + \phi_n x_n + \phi_{n-1} x_{n-1} + \cdots + \phi_1 x_1 + \varepsilon_{n+1}$$

式中：ξ_{n+1} 表示在 n+1 时间的随机误差，另外，时间序列中的每个元素都可以看作是随机误差与前面元素值的线性组合的结合；φ_i 是自回归参数。或者也可将序列中的值看作是前面的元素值与均值的偏差的加权和。

自回归模型可以是非平稳的也可以是平稳的。

时间序列元素值之间的另一个依赖关系是移动平均。此时，未来值可以通过对前面的一组连续值应用移动平均得到，即

$$x_{n+1} = a_{n+1} + \theta_n a_n + \theta_{n-1} a_{n-1} + \cdots + \theta_{n-q} a_{n-q}$$

其中：a_i（i=n+1,⋯,n-q）表示一种冲击。可以应用很多不同的移动平均模型进行时间序列未来值的预测。但是要注意，应用移动平均的点与预测值之间可能会存在一个时滞。也可以同时应用自回归和移动平均来对时间序列预测，这种方法称为自回归移动平均模型。

23.7　例题

例 4.51　一般来说，随着技术的进步和生产的增长，新产品的增长在其未达饱和之前遵循指数曲线增长规律，但随着产品销售量的增加，产品总量会接近于社会饱和量，此时预测模型应用修正指数曲线。

表 23.1 是某厂收音机销售量统计表，请预测下一年的销售量。

表 23.1　某厂收音机销售量

时间（年）	1969	1970	1971	1972	1973	1974	1975	1976	1977
销售量（万部）	42.1	47.5	52.7	57.7	62.5	67.1	71.5	75.7	79.8
时间（年）	1978	1979	1980	1981	1982	1983			
销售量（万部）	83.7	87.5	91.1	94.6	97.9	101.1			

解：

计算前应对数据进行检验，看给定数据的逐期增长量的比率$\frac{y_{t+1}-y_t}{y_t-y_{t-1}}$是否接近某一常数，如果是，则可以采用修正指数曲线。对于本例而言，此比例落在[0.9429,0.9762]区间，可认为是一个常数，因此本例的预测可以采用修正指数曲线模型。

```
>> x=[42.1000 47.5000 52.7000 57.7000 62.5000 67.1000 71.5000 75.7000 79.8000
      83.700087.5000 91.1000 94.6000 97.9000 101.1000];
>> y=expcurve(x,1)
 y= range: [0.9429 0.9762]   %比率值
val: 104.2037
a: -143.2063
b: 0.9608
k: 179.7162
```

例 4.52 Gomperta 曲线是一种常用的时间序列模型。它的特点是开始增长很慢，随后逐渐加快，同时达到一定阶段变慢直到增长速度慢慢趋于 0。其走向很像一个顺时针倾斜的字母 S。该模型的数学表达式为

$$y = ka^{b^t}$$

请用此模型对例 4.51 中的数据进行预测。

解：

```
>> x=[42.1000 47.5000 52.7000 57.7000 62.5000 67.1000 71.5000 75.7000 79.8000
      83.7000 87.5000 91.1000 94.6000 97.9000 101.1000];
>> y=gomperta(x,1)
 y= range: [0.9429 0.9762]
val: 103.4399
a: 0.2840
b: 0.9048
k: 133.3341
```

例 4.53 Logistic 曲线是数学家 Veihulot 在研究人口增长规律时首先提出的。它的特点与 Gomperta 曲线相似。在很多情况下，这两种模型是可以互换使用的。

Logistic 曲线的数学式为

$$y = \frac{k}{1+me^{-at}} \text{ 或 } y = \frac{1}{k+ab^t}$$

在使用此模型时，K 值既可以指定也可能通过计算而得。

请用此模型，对下列时间序列数据进行分析。

```
y=[41 51 71 166 248 329 360 381 399]
```

解：

```
>> x=[41 51 71 166 248 329 360 381 399];
>> y=logistic_curve(x,1,410)   %指定k
 y=range: [0.2471 1.4593]
val: 404.1176
m: 29.0654
a: 0.7599
>> y=logistic_curve(x,1)   %不指定k
 y= range: [0.2471 1.4593]
val: 407.1473
a: 0.7201
m: 25.1414
k: 414.7849
```

例 4.54　在时间序列的预测中，由于自适应滤波法的预测模型简单，又可以在计算机上对数据进行处理，所以这种预测方法应用较为广泛。

自适应滤波技术有两个明显的优点：一是技术比较简单，可根据预测意图来选择权数的个数和学习常数，以控制预测，也可以由计算机自动选定。二是它使用了全部历史数据来寻求最佳权系数，并随数据轨迹的变化而不断更新权数，从而不断改进预测。

下面利用此技术预测 x=0.1:0.1:1 时间序列。

解：

自适应滤波法的基本预测公式为

$$\hat{y}_{t+1} = w_1 y_t + w_2 y_{t-1} + \cdots + w_N y_{t-N+1} = \sum_{i=1}^{N} w_i y_{t-i+1}$$

式中：w_i为权重；N为权重个数；$\hat{y}_{t+1}$为第t+1期的预测值；权重的修正公式为：

$$w_i = w_i + 2ke_{i+1}y_{t-i+1}$$

式中：k为学习常数；e_{i+1}为第t+1期的预测值的误差；$t=N,N+1,\cdots,n$，n为序列长度。可以看出，权重的调整项包括预测误差、原观测值和学习常数三个因素，学习常数k的大小决定权数调整的速度。

根据自适应滤波法的原理，可编程计算，得到结果如下：

```
>> x=0.1:0.1:1;
>> y=adapt_curve(x,2,[3 4 7 9],0.9)
 y=w: [1.9999 -0.9999]                    %最后的权重
val: [1.3000 1.4000 1.7001 1.9003]        %预测值
```

例 4.55　均生函数是一种较为常用的时间序列预测技术。表 23.2 为 1950—1980 年期间我国收入指数，求其延拓均生函数，并分析优势周期。

表 23.2　1950—1980 年我国收入指数

年　份	0	1	2	3	4	5	6	7	8	9
1950	119.0	116.7	122.2	114.0	105.8	106.4	114.1	104.5	122.0	108.2
1960	98.6	70.3	93.5	110.7	116.5	117.0	117.0	92.8	93.6	119.3
1970	123.3	107.0	102.9	108.3	101.1	108.3	97.3	107.8	112.3	107.0
1980	106.4	104.9	108.3	109.8	113.5	112.7				

解：

对有时间序列 $x(t)=\{x(1),x(2),\cdots,x(N)\}$，定义均值生成函数

$$\bar{x}_l(i)=\frac{1}{n_l}\sum_{j=0}^{n_l-1}x(i+jl)\ i=1,\cdots,l\ 1\leqslant l\leqslant M$$

式中：n_l 为满足 $n_l\leqslant[\frac{N}{l}]$ 的最大整数，$M=[\frac{N}{2}]$ 为不超过 N/2 的最大整数，当 N 为偶数时，$[\frac{N}{2}]=\frac{N}{2}$，当 N 为奇数时，$[\frac{N}{2}]=\frac{N-1}{2}$。

由此可见，均值生成函数是由时间序列按一定的时间间隔计算均值而派生出来的。

对均生函数作周期性延拓，即构造如下的矩阵

$$F=\begin{vmatrix}\bar{x} & \bar{x}_2(1) & \vdots & \bar{x}_M(1)\\ \bar{x} & \bar{x}_2(2) & \vdots & \bar{x}_M(2)\\ \vdots & \vdots & \vdots & \vdots\\ \bar{x} & \bar{x}_2(i_2) & \vdots & \bar{x}_M(i_M)\end{vmatrix}$$

其中：$\bar{x}_2(i_2)$ 表示取 $\bar{x}_2(1)$，$\bar{x}_2(2)$ 之一，其余类推。称 f_l，$l=1,2,\cdots,M$ 为延拓均生函数。

根据均生函数的定义，可编程计算，得到如下结果。

```
>> x=[119.0 116.7 122.2 114.0 105.8 106.4 114.1 104.5 122.0 108.2...
       98.6   70.3  93.5 110.7 116.5 117.0 117.0   92.8  93.6 119.3...
      123.3 107.0 102.9 108.3 101.1 108.3  97.3  107.8 112.3 107.0...
      106.4 104.9 108.3 109.8 113.5 112.7];
 >>y=meangcyc(x)
   y=13  8  5  7  17         %优势周期
```

例 4.56　某气象站 1958—1977 年 7 月降水量（mm）序列为：$x(t)$=[130 50 220 140 100 380 110 140 110 220 160 170 410 70 60 200 170 70 220 190]，请预报 1978 年的降水量。

解：

利用均生函数求得的优势周期就可以预测时间序列。通过计算得到预测值为 90mm，实际值为 100mm。

```
 >> x=[130 50 220 140 100 380 110 140 110 220 160 170 410 70 60 200 170 70 220
190];
```

```
>> [y,L]=meangcyc(x,1)
 y=90            %预测值
  L=7            %优势周期
```

例 4.57　赤道东太平洋地区是一个反映全球大气和海洋变化的敏感区域，对全球气候有着重大影响的厄尔尼诺现象就发生在这里。表 23.3 给出了这一地区 1951—1985 年 35 年秋季（9—11 月）海温的观察值，试用逐步回归方程及主成分分析建立预测模型。

表 23.3　海温观察值（℃）

海　温	26.6	25.7	25.2	24.7	25.4	26.6	26.0	25.7	25.9	25.5	25.4	26.4	24.9	26.8	25.4	25.1	26.0
	26.5	25.2	25.1	27.2	24.9	25.3	24.8	26.6	25.9	25.4	25.9	25.7	25.5	26.7	26.1	25.3	25.4

解：

（1）逐步回归法。

```
>>load mydata;
>> y=meang(x1);
>> y1=[ones(34,1) y(:,2:17)]; stepwise(y1,x1')      %逐步回归工具
>> beta;                                    %回归系数，其中 fi 为第 i 个延拓均生函数
```

从逐步回归过程中可得到最终的回归方程为

$$\hat{x}(t) = -0.9732f_2 - 0.9817f_3 + 0.8023f_6 + 0.2382f_7 + 0.4007f_{10} + 0.4388f_{11} + 0.3757f_{15} + 0.4228f_{16} + 0.2633f_{17}$$

根据回归方程可得到图 23.2。

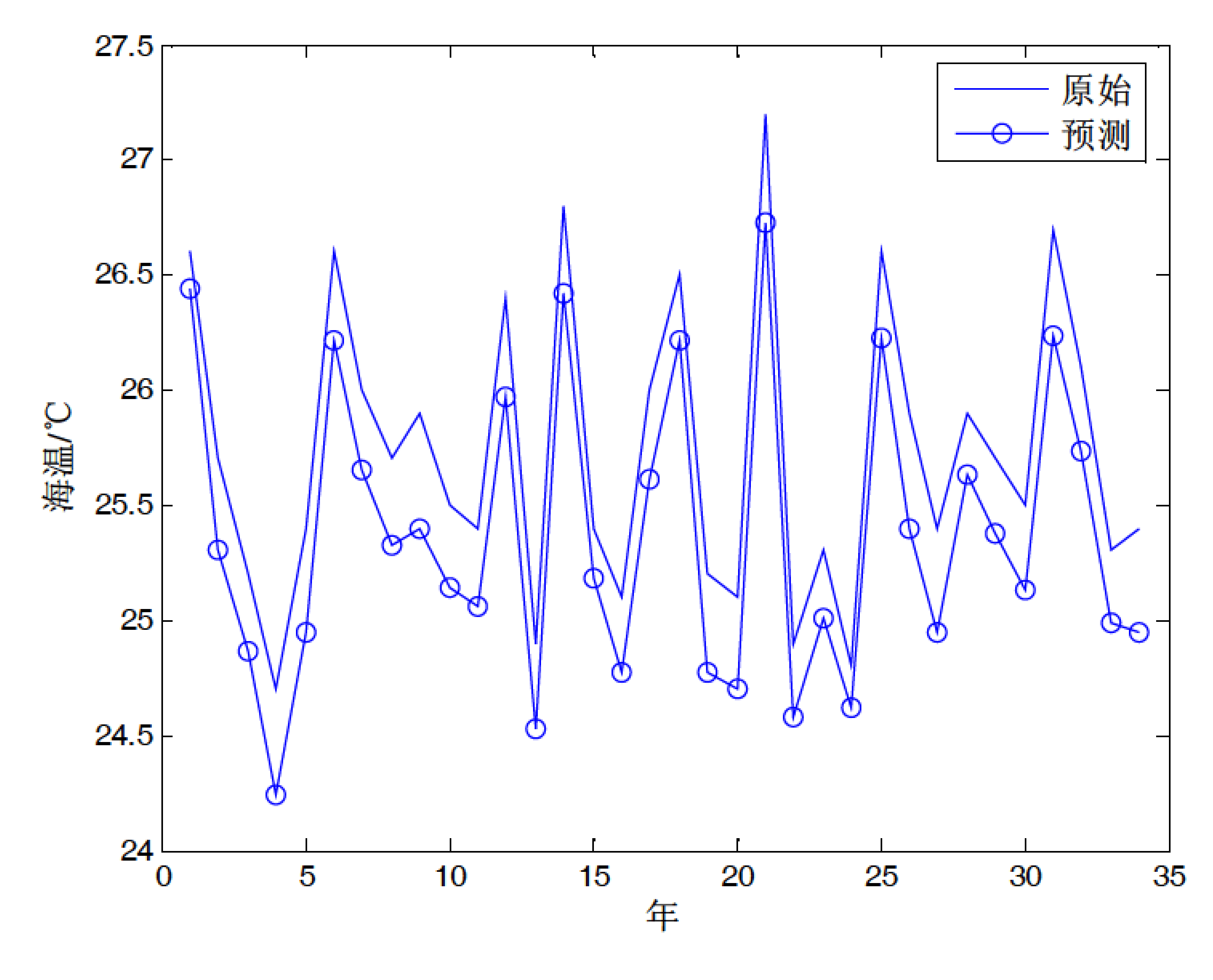


图 23.2　预测与实际值

如果要对以后的时间点进行预测，则可以对回归方程中的各延拓均生函数进行延拓，然后再根据回归方程式计算即可。如要预测下一个时间点的海温值，则其值为

```
yn=-0.9732*25.7765-0.9817*25.7364+0.8023*25.56+0.2382*26.475+0.4007*25.8…
    +0.4388*25.1333+0.3757*25.25+0.4228*25.2+0.2633*26.55
=24.9602
```

（2）主成分分析法。

```
>> y=meangprin(x);
>> plot(1:34,x,'o-'); hold on; plot(1:34,y,'o-');  %图 23.3
```

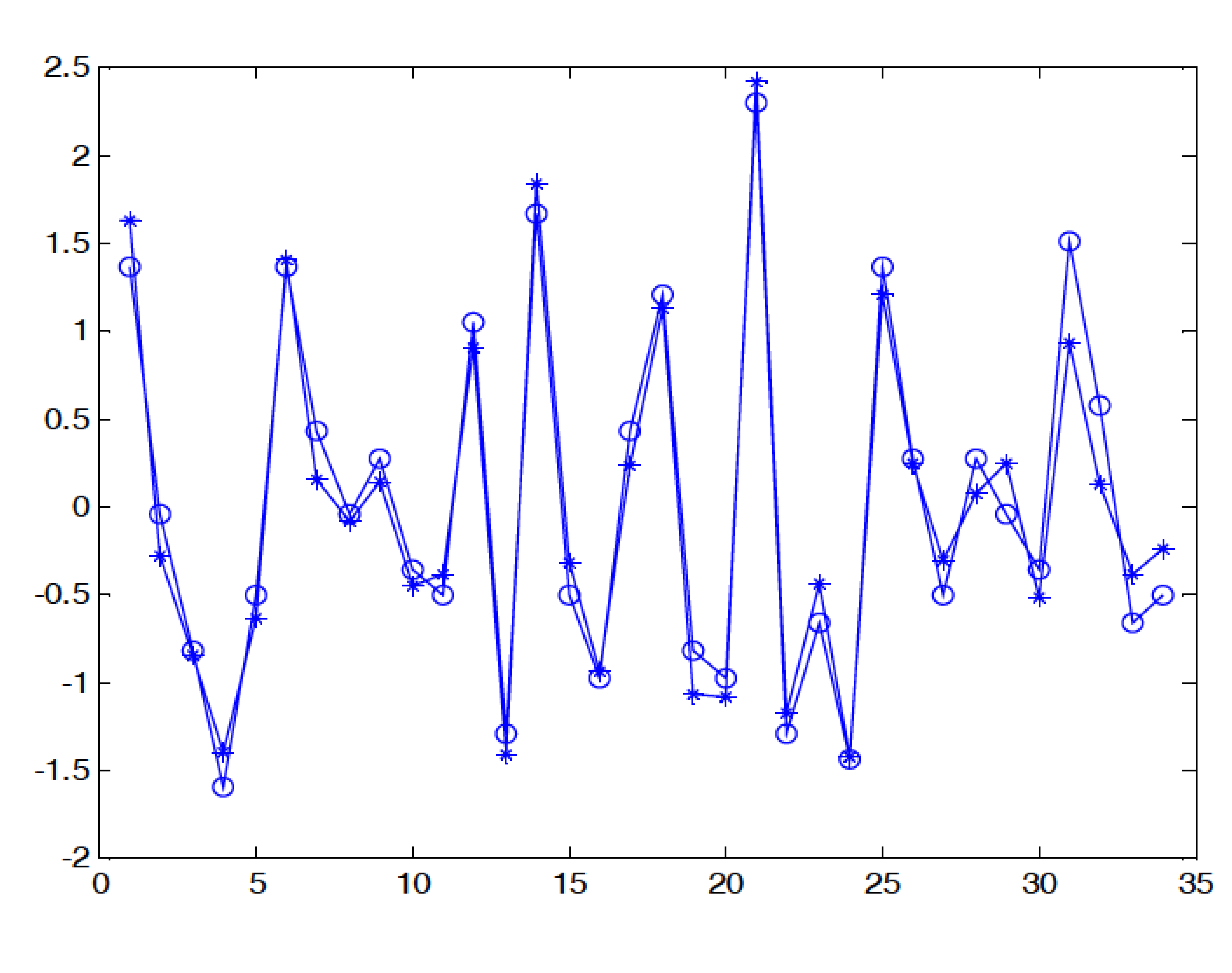


图 23.3 预测与实际值

从图中可看出，此结果要好于第 1 种方法。

例 4.58 铁路客运量预测是铁路旅客运输组织工作的重要基础，能为合理安排运输计划提供重要的决策支持，有助于铁路旅客运输企业根据客运市场的变化，动态实时地调整产品结构，对促进铁路客运的发展具有十分重要的意义。

从预测时间的长短角度考虑，可将铁路客运量预测方法分为长期预测研究和短期预测研究。前者是根据影响铁路客运量因素（如流动人口、GDP、铁路基础设施建设情况等）来预测今后较长一段时期内的客运量；而后者则是从短期时间内比如月、日的客运量变化等为出发点，研究铁路客运量的变化情况。

表 23.4 是我国某段时间内铁路客运量及相关因素值，请预测下一年的客运量（即表中空格中值）。

解：

可以用多种方法求解本问题。在此例中采用多元线性回归方法。

```
>>y=[933089508510016410507310515510560697260111764115583125656135670]';
>>x=[78973 123626 6.60;84402.3 124761 6.64;89677.1 125786 6.74;99214.6 126583
```

```
6.87;109655.2 127627 7.01;120332.7 128453 7.19;135822.8 129227 7.30;
159878.3 129988 7.44;183217.4 130756 7.54;211923.5 131448 7.71;257305.6
132129 7.80];
>>xx=[300670.0 132803 7.97];
>> [beta,stats,yy,ylr]=myregress(x,y,xx,0.95,'m');  %0.95 为置信度
```

表 23.4　客运量及其影响因素

铁路客运量（万人）	GDP（亿元）	人口（十万人）	铁路营业里程（万公里）
93308	78973	123626	6.60
95085	84402.3	124761	6.64
100164	89677.1	125786	6.74
105073	99214.6	126583	6.87
105155	109655.2	127627	7.01
105606	120332.7	128453	7.19
97260	135822.8	129227	7.30
111764	159878.3	129988	7.44
115583	183217.4	130756	7.54
125656	211923.5	131448	7.71
135670	257305.6	132129	7.80
	300670.0	132803	7.97

从计算结果可看出，第 7 个样本预测误差较大（图 23.4），所以应采用稳健回归方法。

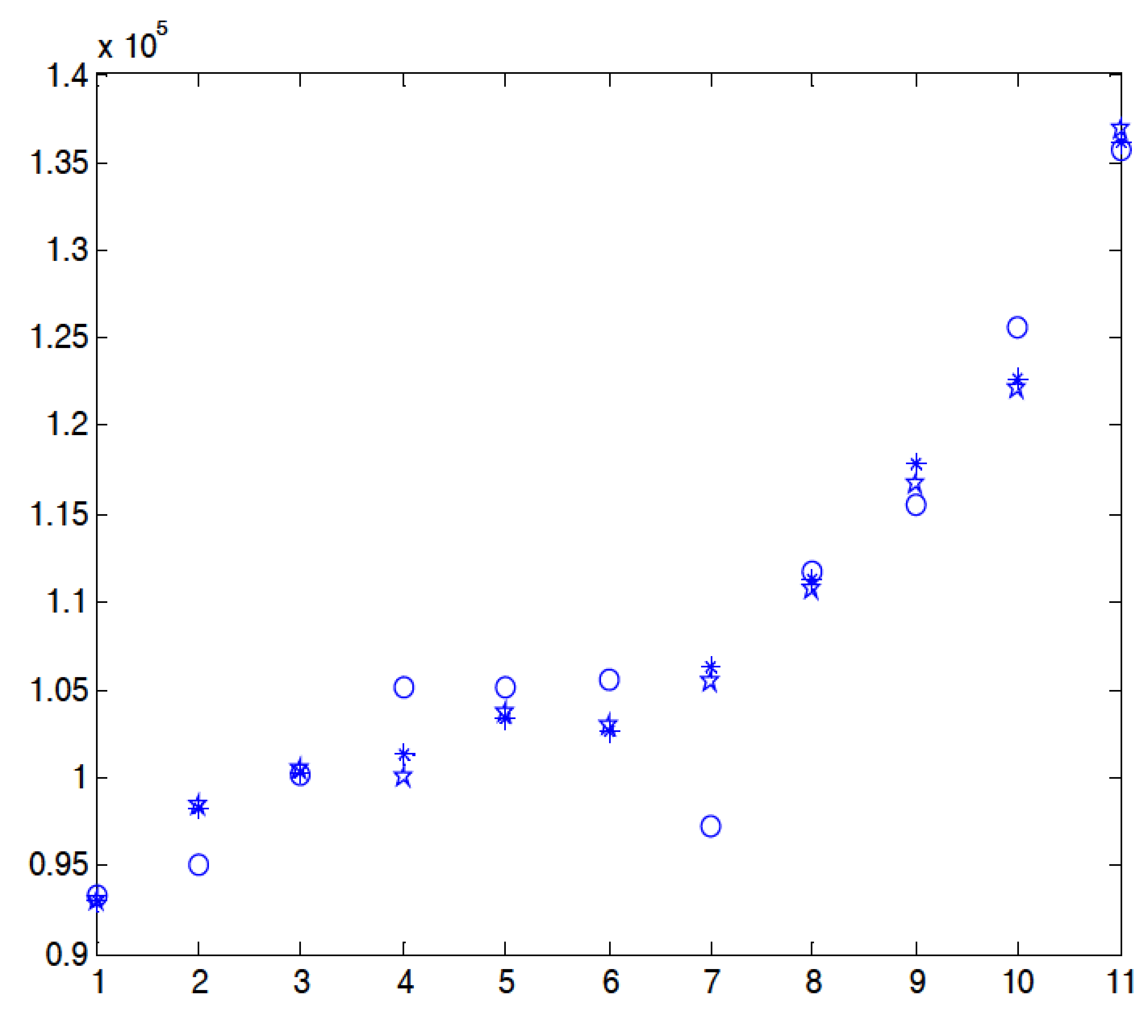


图 23.4　客运量的实际值及不同方法的预测值

下一年的预测结果为 146110，实际值为 143754。

例 4.59　对由下式产生的时间序列（长度为 1000）进行 1、2、3 步的预测。

$$X_t - 0.8X_{t-1} = \varepsilon_t - 0.4\varepsilon_{t-1} \qquad \varepsilon_t \sim N(0,1)$$

解：

根据 Garch Toolbox 中自回归的相关函数，可计算如下。从结果可看出，两者的结果相关较差。这主要是由于没有对模型优化，即对自回归模型中的阶数没有优化。

```
>>randn('state',sum(clock));
 epls=randn(1,1000);x(1)=0;
 for j=2:1000
    x(j)=0.8*x(j-1)+epls(j)-0.4*epls(j-1);
 end
 spec=garchset('R',1,'M',1,'display','off');                    %设定模型
 [coeffx,errorx,LLfx]=garchfit(spec,x);                          %模拟
  [sigmaforcast,x_forcast]=garchpred(coeffx,x,3);                %预测
  x_theory(1)=0.8*x(1000);x_theory(2)=0.8*x_theory(1);x_theory(3)=0.8*x_t
heory(2);
```

也可以利用以下 MATLAB 中的相关函数进行计算，得到图 23.5。

```
  >> th=ivar(x',5);a=th2arx(th);data1=predict(x(1:50)',th);    %预测
  >> e1=pe(x(1:50)',th);plot(e1);hold
on;plot(x(1:50),'*-');plot(data1{1},'p-');
```

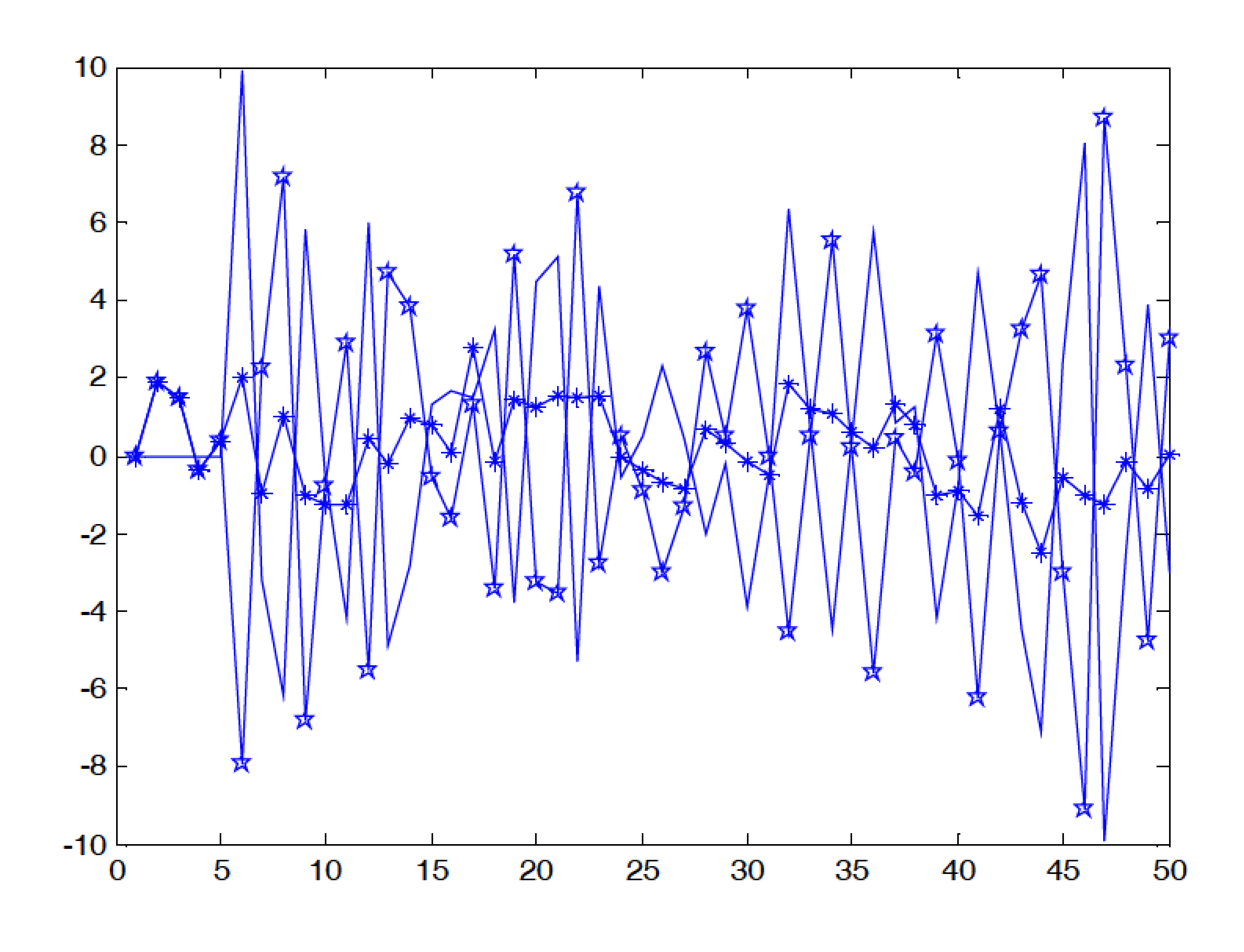


图 23.5　原始信号、模拟信号及误差图

例 4.60　表 23.5 为某水井水位值数据。请预报下一年每个月（即 12 个）的水位值。

下一年的预测结果为 146110，实际值为 143754。

例 4.59　对由下式产生的时间序列（长度为 1000）进行 1、2、3 步的预测。

$$X_t - 0.8X_{t-1} = \varepsilon_t - 0.4\varepsilon_{t-1} \qquad \varepsilon_t \sim N(0,1)$$

解：

根据 Garch Toolbox 中自回归的相关函数，可计算如下。从结果可看出，两者的结果相关较差。这主要是由于没有对模型优化，即对自回归模型中的阶数没有优化。

```
>>randn('state',sum(clock));
 epls=randn(1,1000);x(1)=0;
 for j=2:1000
    x(j)=0.8*x(j-1)+epls(j)-0.4*epls(j-1);
 end
 spec=garchset('R',1,'M',1,'display','off');                    %设定模型
 [coeffx,errorx,LLfx]=garchfit(spec,x);                         %模拟
  [sigmaforcast,x_forcast]=garchpred(coeffx,x,3);               %预测
  x_theory(1)=0.8*x(1000);x_theory(2)=0.8*x_theory(1);x_theory(3)=0.8*x_t
heory(2);
```

也可以利用以下 MATLAB 中的相关函数进行计算，得到图 23.5。

```
  >> th=ivar(x',5);a=th2arx(th);data1=predict(x(1:50)',th);     %预测
  >> e1=pe(x(1:50)',th);plot(e1);hold
on;plot(x(1:50),'*-');plot(data1{1},'p-');
```

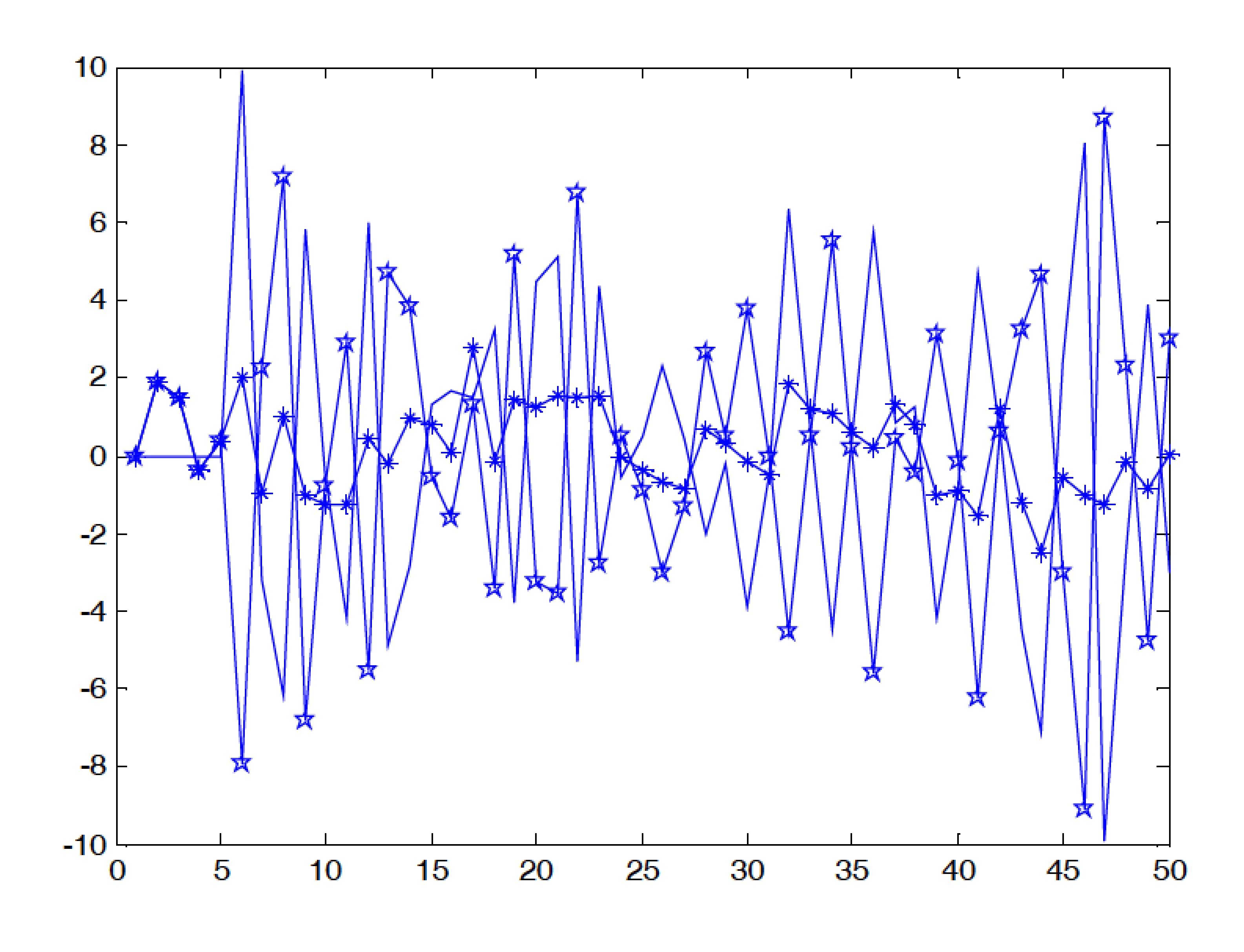


图 23.5　原始信号、模拟信号及误差图

例 4.60　表 23.5 为某水井水位值数据。请预报下一年每个月（即 12 个）的水位值。

下一年的预测结果为 146110，实际值为 143754。

例 4.59　对由下式产生的时间序列（长度为 1000）进行 1、2、3 步的预测。

$$X_t - 0.8X_{t-1} = \varepsilon_t - 0.4\varepsilon_{t-1} \qquad \varepsilon_t \sim N(0,1)$$

解：

根据 Garch Toolbox 中自回归的相关函数，可计算如下。从结果可看出，两者的结果相关较差。这主要是由于没有对模型优化，即对自回归模型中的阶数没有优化。

```
>>randn('state',sum(clock));
 epls=randn(1,1000);x(1)=0;
 for j=2:1000
    x(j)=0.8*x(j-1)+epls(j)-0.4*epls(j-1);
 end
 spec=garchset('R',1,'M',1,'display','off');                    %设定模型
 [coeffx,errorx,LLfx]=garchfit(spec,x);                          %模拟
  [sigmaforcast,x_forcast]=garchpred(coeffx,x,3);                %预测
  x_theory(1)=0.8*x(1000);x_theory(2)=0.8*x_theory(1);x_theory(3)=0.8*x_t
heory(2);
```

也可以利用以下 MATLAB 中的相关函数进行计算，得到图 23.5。

```
  >> th=ivar(x',5);a=th2arx(th);data1=predict(x(1:50)',th);     %预测
  >> e1=pe(x(1:50)',th);plot(e1);hold
on;plot(x(1:50),'*-');plot(data1{1},'p-');
```

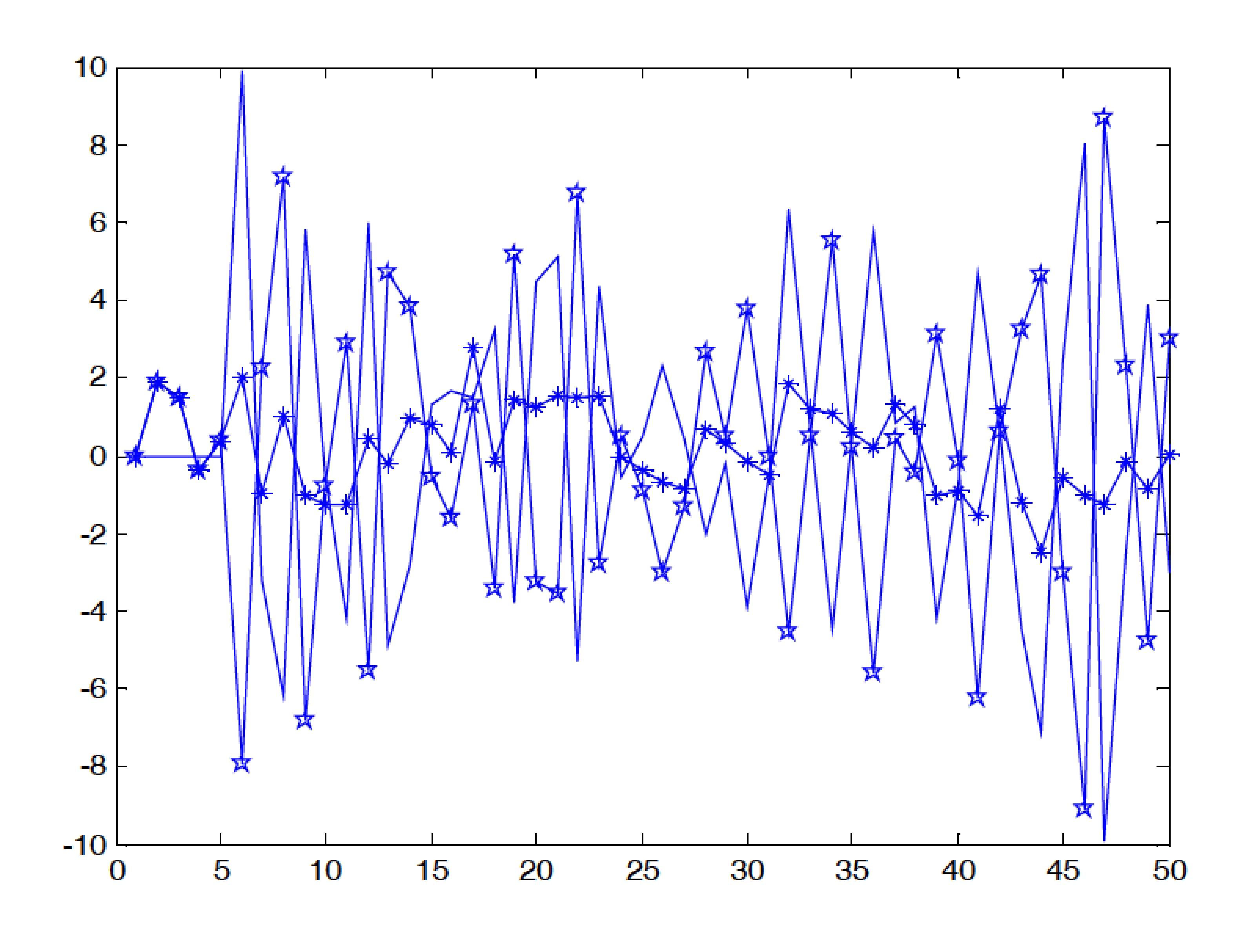


图 23.5　原始信号、模拟信号及误差图

例 4.60　表 23.5 为某水井水位值数据。请预报下一年每个月（即 12 个）的水位值。

表 23.5　水井水位值　　单位：米

年　次	1	2	3	4	5	6	7	8	9	10	11	12
1	9.40	8.81	8.65	10.01	11.07	11.54	12.73	12.43	11.64	11.39	11.0	10.85
2	10.71	10.24	8.48	9.88	10.31	10.53	9.55	6.51	7.75	7.80	5.95	5.21
3	6.39	6.38	6.51	7.14	7.26	8.49	9.39	9.71	9.65	9.26	8.84	8.29
4	7.21	6.93	7.21	7.82	8.59	9.59	8.77	8.61	8.94	8.81	8.50	8.30
5	7.66	7.68	7.85	8.53	9.38	10.09	10.59	10.83	10.49	9.21	8.66	8.39
6	8.27	8.14	8.71	10.43	11.47	11.73	11.61	11.93	11.55	11.35	11.11	10.4
7	10.16	9.96	10.47	11.70	10.1	10.37	12.47	11.91	10.83	10.64	10.29	10.34

解：

在处理时间序列时，首先要判断序列是否平稳。这可以从原信号图、自相关系数图和偏相关系数图 23.6 看出，只有当 3 个图的基线平稳，才可以认为是平稳信号。很明显，此例中的信号不是一个平稳序列，另外还有明显的季节性。

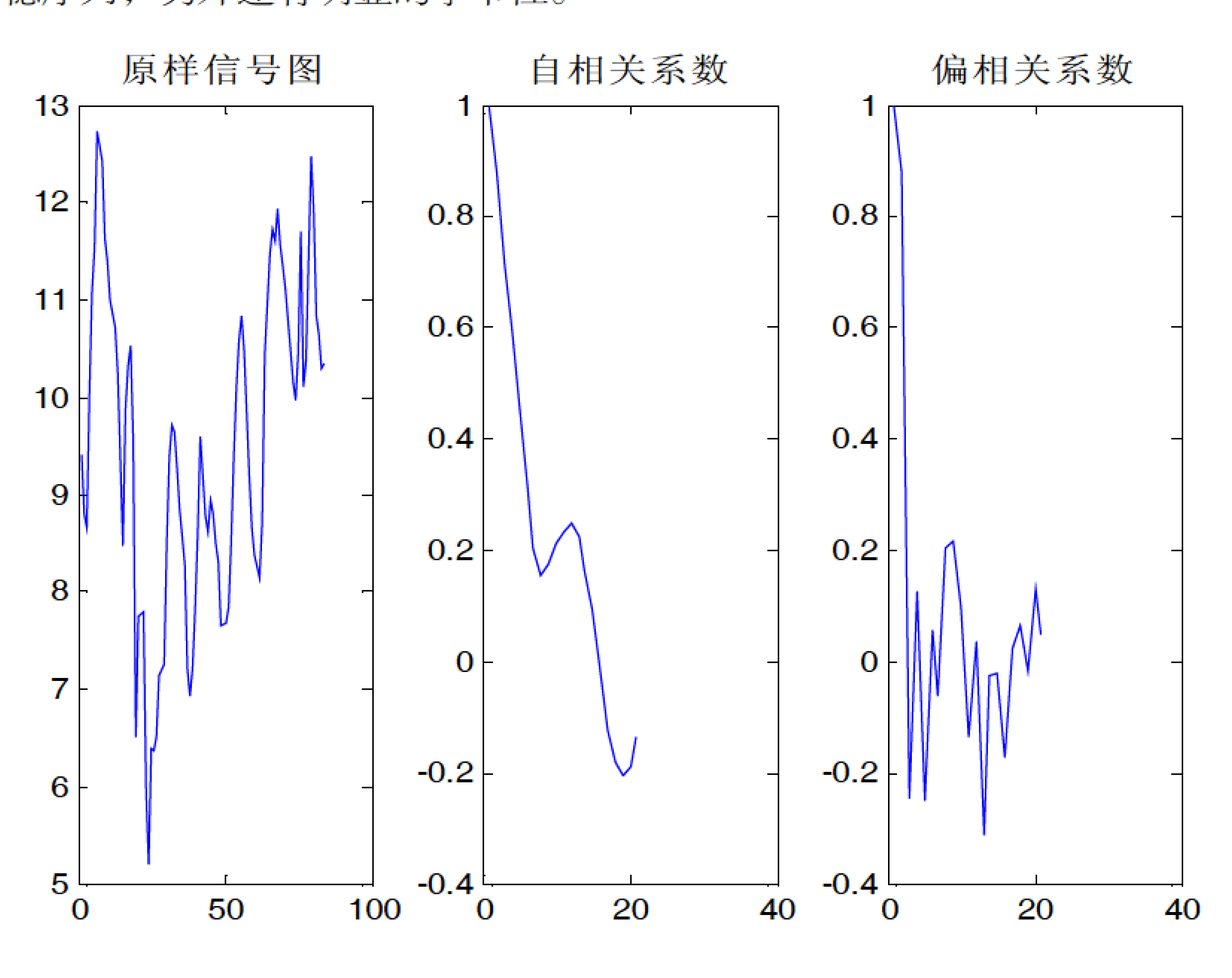


图 23.6　信号、自相关和偏相关系数图

要使非平稳信号变成平稳信号，最简便的方法便是对原信号作差分运算。至于要作多少次差分，可以从自相关系数的平稳性及差分后的信号稳定性作出判断。在此例中作一次差分即可。

```
>>load data;
>> [x_forcast,coeff,order]=myar(x,12);
   x_forcast =[10.0970  8.7265  9.5783  11.6732  10.9280  11.3674  12.3747
             11.5573   11.3022   10.6489    9.2565    9.3095];     %预测结果
 order=15     6                                                  %模型的阶数
```

例 4.61　时间序列中相似子序列的搜索是时间序列数据挖掘中的一项很有意义的工作。一般常使用欧氏距离作为对象间相似性的度量，距离越小它们越相似。但对时间序列而言，因有可能存在空间弯曲现象，所以欧氏距离不适合作为时间序列相似性搜索度量，而要使用动态距离 dtw（或称弯曲距离）。

现随机产生 3 个时间序列，并求它们之间的动态弯曲距离。

解：

根据算法原理，可编程计算得到如下的结果。

```
>>x=1+3*rand(1,300);x1=1+2*rand(1,300);y=1+rand(1,100);
 >> [dist,w]=mydtw(y,x1);        %函数输出变量中的 w 为路径
    dist= 8.5341
 >> [dist,w]=mydtw(y,x);
  dist=17.3455
```

例 4.62　在解决时间序列的实际问题时，为了减少计算量，常常需要压缩，以减少它的长度，但不失真。压缩的方法有很多，分段线性表示其中的一种。

试对下列序列用分段线性方法表示。

```
t=[10.5 10.44 9.94 10.25 11.0 9.88 10.5 12.0 13.94 12.25 12.61 13.5 13.44 12.44
13.5 15.39 15.75 13.88 14.5 15.5 16.13 14.75 11.75 15.25 17.13 20.3 19.0 21.5];
```

解：

分段线性表示时间序列有不同的分割方法，函数中采用3种方法。图23.7为计算结果。

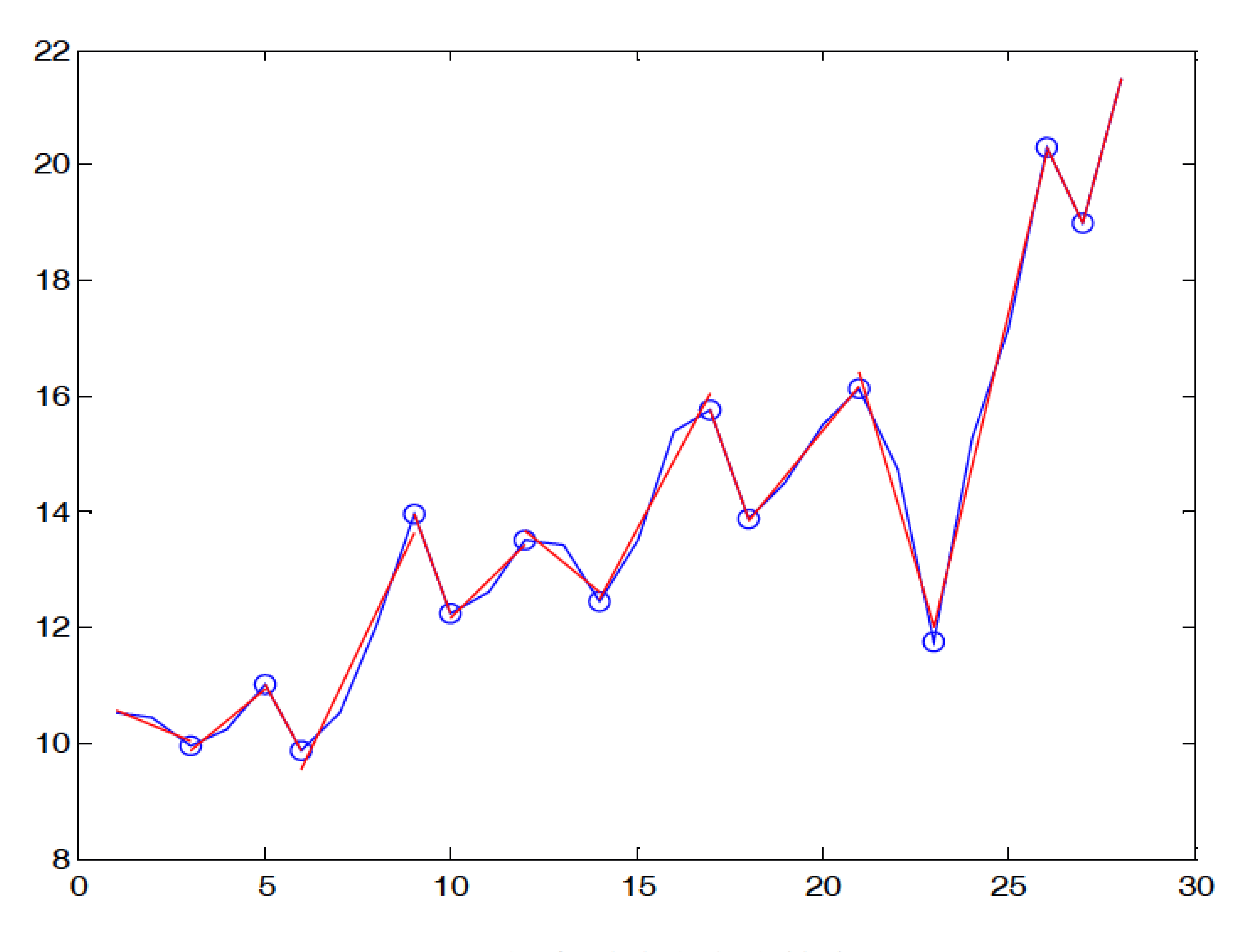


图 23.7　原序列的分段线性表示

可看出用15个点来代表原来28个点所组成的序列。y中第1列为分割点，第2～23列为每段线的斜率和截距，第4列为原序列每段的误差，用每点残差平方和表示。

```
>>[y,me,lamda]=PLR(x,1,0.15);     %0.15为阈值，这个值直接影响结果
```

例 4.63　时间序列的异常检测是时间序列挖掘的一个重要内容，在网络入侵、故障检测等领域中有着广泛的应用。

异常点有检测有很多种方法，最简单的便是根据统计学原理，其值与均值的差异超过 2 倍的标准差的点有可能是异常点。现根据这一原理，对某一时间序列（图 23.8）找出可能的异常点。

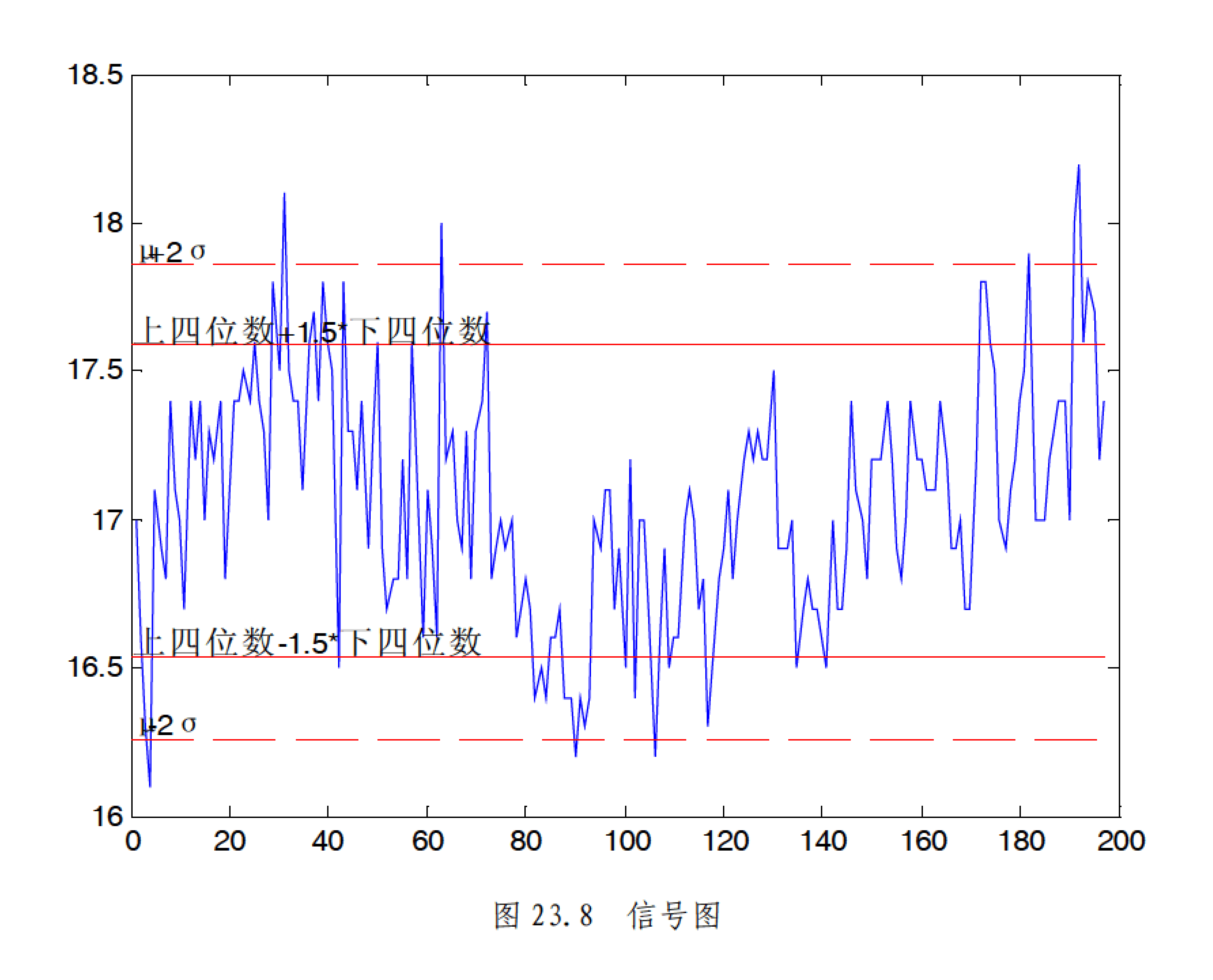


图 23.8　信号图

解：

```
>>load data;
>>n=length(x);plot(x);hold on;
 a1=prctile(x,75);a2=prctile(x,25);R1=a1-a2;a1=a1-1.5*R1;a2=a2+1.5*R1;
 line([0 n],[a1 a1],'color','r');text(0.04,a1+0.04, '上四位数-1.5*下四位数')
 line([0 n],[a2 a2],'color','r');text(0.02,a2+0.04,'上四位数+1.5*下四位数')
 c0=mean(x);cs=std(x);c1=c0+2*cs;c2=c0-2*cs;
 line([0 n],[c1 c1],'color','r','linestyle','--');text(0.4,c1+0.04,'μ+2σ');
 line([0 n],[c2 c2],'color','r','linestyle','--');text(0.4,c2+0.04,'μ-2σ');
 U=x-c0;y=find(U>2*cs);y=[y find(U<-2*cs)];
```

得到以下的点为异常点的可能性较大：

```
 y=31  63  182  191  192  4  90  106
```

例 4.64　在时间序列的数据挖掘中，基于小波分析的技术也是常用的一种方法。小波分析既可以将时间序列降噪、降维，也可以进行相似性的应用。

下面对一系列的模拟信号进行小波处理。

解：

图23.9为两个随机产生的信号图，从图中可直观看出这两者间相似度较小。

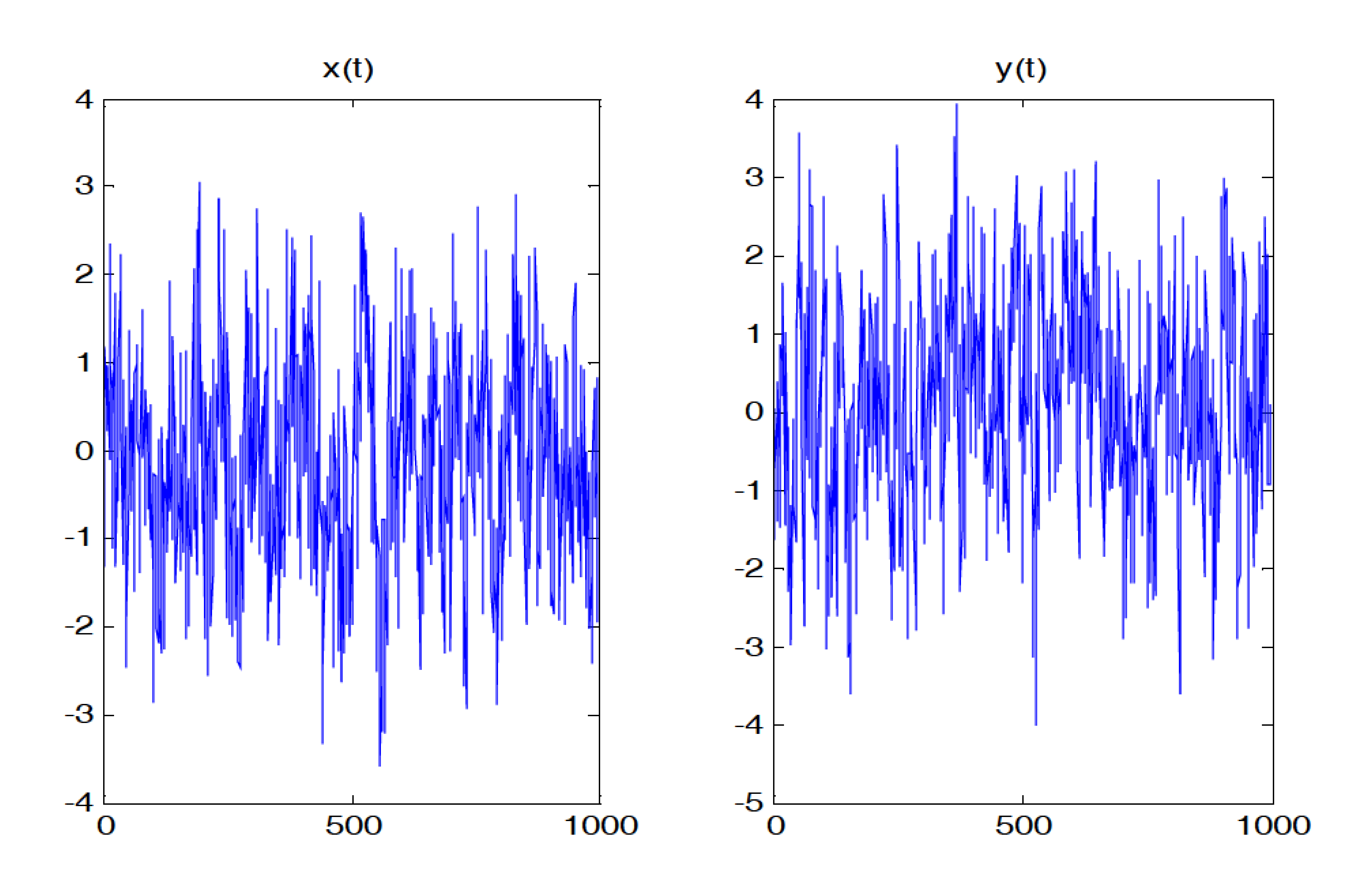


图 23.9　原始信号图

对它们进行相似度分析可得到如下结果。

```
>> epls=randn(1,1000);x(1)=0;for j=2:1000;x(j)=0.8*x(j-1)+epls(j)-0.4*epls(j-1);end
>> epls=randn(1,1000);y(1)=0;for j=2:1000y(j)=0.8*y(j-1)+epls(j)-0.4*epls(j-1);end
>>subplot(121),plot(x);subplot(122),plot(y);
>> [Seqsim,Sim]=wavesim(x,y,2,3);
>>Sim=0.5641    %即为相似度
```

例 4.65　小波分析还可以用于对信号的降噪、降维的处理。试用此方法对例 4.64 中的 *x* 原始信号进行相应的处理。

解：

```
>>wname='sym6';lev=5;
>>[c,l]=wavedec(x,lev,wname);
>>sigma=wnoisest(c,l,1);
>> alpha=2;thr=wbmpen(c,l,sigma,alpha);                  %阈值
>>keepapp=1;
>> xd=wdencmp('gbl',c,l,wname,lev,thr,'s',keepapp);      %降噪后的信号
>> plot(xd)                                              %图 23.10
```

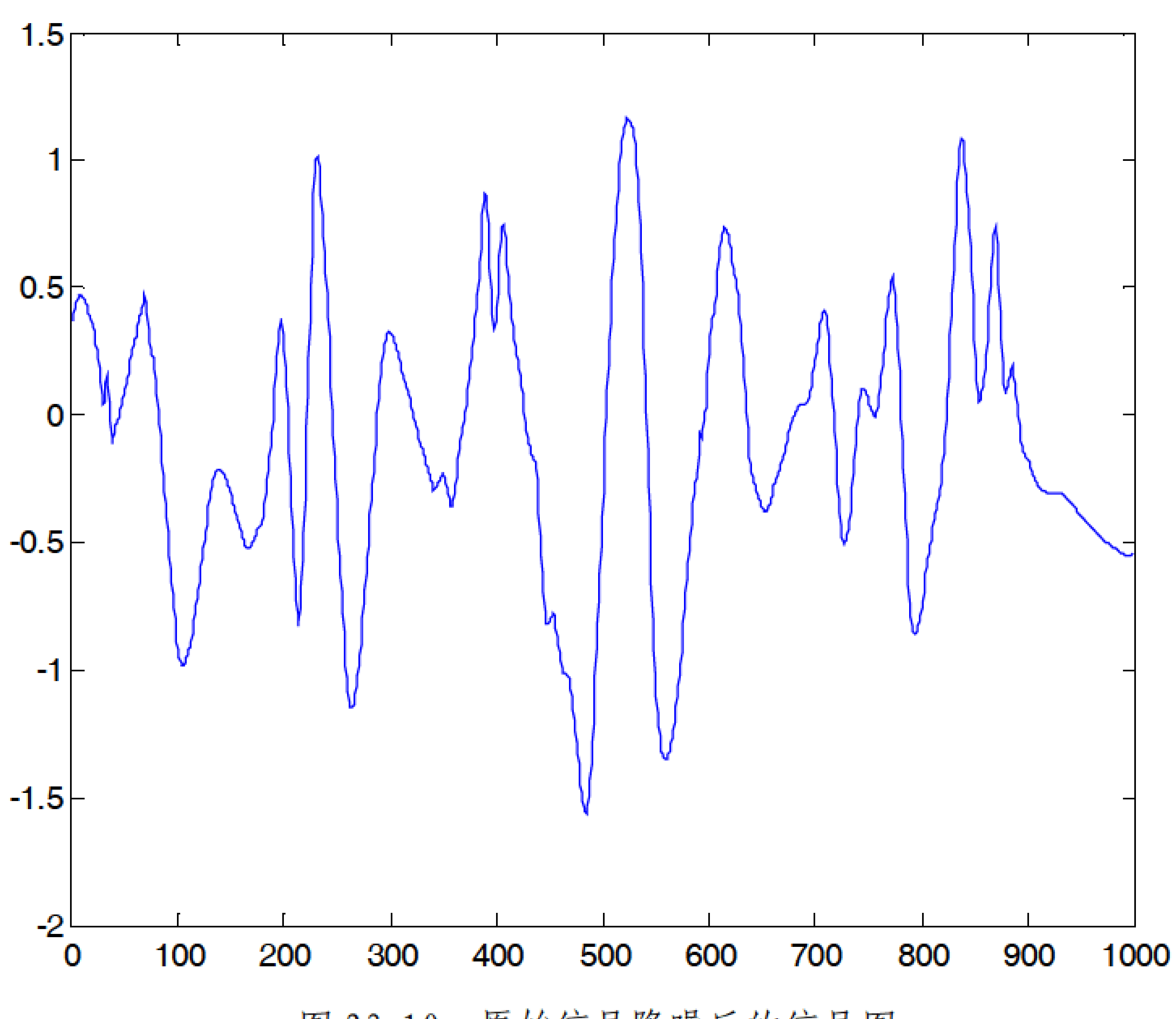


图 23.10　原始信号降噪后的信号图

以下对原始信号进行降维：

```
>> [c,l]=wavedec(x,5,'db3'); d5=wrcoef('d',c,l,'db3',5); d4=wrcoef('d',c,l,'db3',4);
   d3=wrcoef('d',c,l,'db3',3);d2=wrcoef('d',c,l,'db3',2);d1=wrcoef('d',c,l,'db3',1);
```

从图23.11中可看出，适当层的细节系数可代替原始信号。

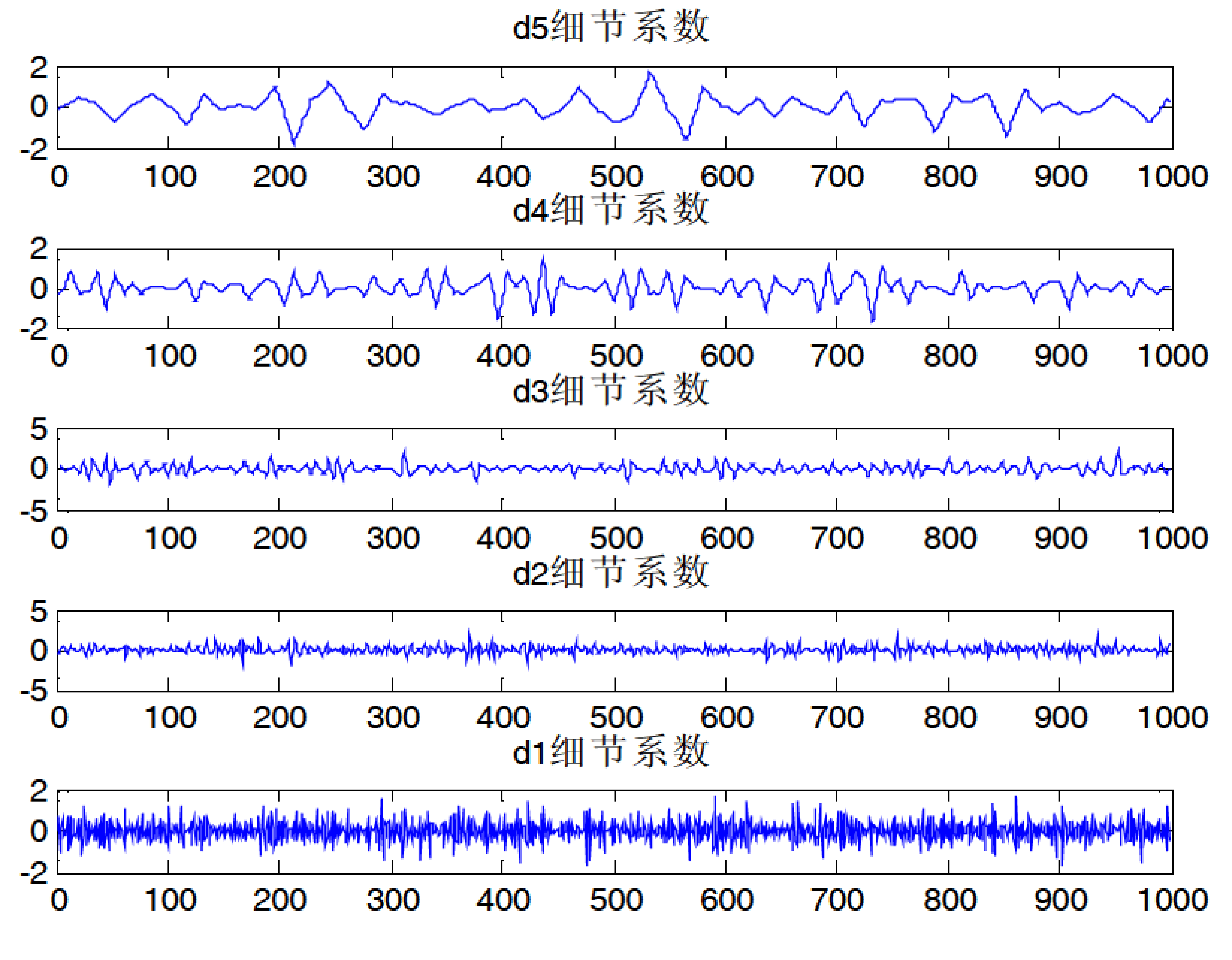


图 23.11　小波分解后的各细节系数图

读书笔记

第 24 章

关联规则挖掘

24.1 关联规则的类型及挖掘算法

关联规则用于发现交易数据库中不同商品（项）之间的联系，这些规则反映了顾客购买的行为模式。发现这样的规则可以应用于顾客购物分析、目录设计、商品广告邮寄分析、商品货架设计、互联网用户的浏览习惯、仓储规划、网络故障分析以及根据购买模式对用户进行分类等方面。

关联规则有许多类型，根据不同的标准，关联规则有不同的分类方法。

根据规则所处理的值的类型可以将关联规则分为布尔关联规则和量化关联规则。若所考虑的关联规则是项的在与不在，则它是布尔关联规则，它表明了离散（分类）对象之间的关系。如果规则所描述的是量化的项或属性之间的关联，则它是量化关联规则，在此规则中，项和属性的量化值划分为区间，涉及动态离散化的数值属性，也可能涉及分类属性。

根据规则中涉及的数据维，可以将关联规则分为单维关联规则和多维关联规则。单维关联规则中的项或属性只涉及单个维或谓词（即一个属性或列），它表明了属性的内在联系。或关联规则涉及两个或多个（不同的）谓词或维，则它是多维关联规则，它表明了属性间的联系，即属性/维之间的关联。

根据规则所涉及的抽象层可以将关联规则分为单层关联规则和多层关联规则。单层关联规则是指在给定的规则集中，规则挖掘只涉及相同抽象层的项或属性。若在给定的规则集中，所挖掘的规则涉及不同的抽象层，则称多层关联规则。

根据关联规则挖掘的不同扩充，关联规则的挖掘可以分为相关分析、最大大模式（最大模式）和大闭项集挖掘。

根据以上的不同类型的关联规则，发展出了相应的关联规则挖掘算法。这些算法的相关内容已在前面章节中做了介绍，在此主要介绍其他的一些关联规则算法。

24.2 基于组织进化的关联规则挖掘

关联规则算法最基本的算法是 Apriori 算法，现在绝大关联规则挖掘算法都基于该算法的框架。随着关联规则算法研究的发展，提出了一些新的算法，如从遗传算法衍生出来的基于组织进化的关联规则算法等算法。

24.2.1 组织的定义

对数据库的数据可以有不同方法的表示。在此用表来表示，其中表中的每一列表示一个属性，表的每一行表示一条数据。先对原始数据进行预处理，经离散化后得到每个属性的取值，用 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$表示每一条数据，其中 x_i 表示属性。

为计算方便，将组织（若干个体的集合）中所有对象取值均相同的属性称为相同属性组织，用 same_{org} 表示。

由于有的组织可以产生规则，有的组织不能产生规则，所以将组织分成自由态组织、异常态组织和正常态组织。

自由态组织是指包含对象个数为 1 的组织，其集合记为 free。异常态组织是指相同属性集合为空的组织，记为 abnormal；其余的组织为正常组织，记为 normal。

24.2.2　组织适应度的计算

规则的支持度、置信度可以从不同的角度表明规则的性质，规则的支持度越大，说明规则在数据集空间所占的比例越大，规则的普遍意义越好；规则的置信度表示由特征推出类别的正确程度，是对关联规则准确度的衡量。在组织进化算法中，先用组织进化算法筛选出满足最小支持度的规则，然后再采用其他常规算法选出满足要求的规则。算法中的适应度函数为

$$F(X)=\begin{cases}0 & (\text{org}\in\text{free})\\ -1 & (\text{org}\in\text{abnormal})\\ \sup(x) & (\text{org}\in\text{normal})\end{cases}$$

其中：sup(x)为关联规则的支持度。可以看出算法中简单地将每个组织的相同属性集转化成规则。

由于每条规则的支持度是不变的，为了避免频繁计算支持度和提取规则方便，可以用表的形式记录规则结构、支持度、置信度等。在计算组织的适应度时，可以先搜索关联规则表，如果存在，则直接记取组织的适应度；否则，计算规则的支持度，然后再计算适应度，并将该关联规则结构和相应的值添加到关联规则。

24.2.3　组织进化算子

组织适应度算法和传统遗传算法的运行机制完全不同，其进行操作作用在组织上，而不是作用在个体上，而且传统遗传算法中的交叉、变异等算子不能直接应用在该算法中，需要重新定义。

（1）合并算子：随机选取两个组织 org_{p1} 和 org_{p2} 作为父代，将其合并为一个子代组织 org_{c}。

（2）增减算子：随机选择两个组织 org_{p1} 和 org_{p2} 作为父代，然后从 org_{p1} 中选择 m%的对象加入 org_{p2} 中，形成两个子代组织 org_{c1} 和 org_{c2}。

（3）交换算子：从一个种群中随机选择两个组织 org_{p1} 和 org_{p2} 作为父代，然后从 org_{p1} 中随机选择 n%的对象加入 org_{p2} 中，再从 org_{p2} 中随机选择 n%的对象加入 org_{p1} 中，形成两个子代组织 org_{c1} 和 org_{c2}。

（4）组织选择算子：从父代组织和子代组织中，选择出适应度较高的组织，并把组织标记为已进行，然后加入下一代。

24.2.4　算法步骤

基于组织进化的关联规则算法的具体步骤如下。

（1）初始化。将每个对象以自由态加入种群 P_0 中，进化代数 $t=0$。

（2）如果在当前进化代数 t 中，种群 P_t 中未进化的组织个数大于 1，则转到第 3 步，否则转到第 5 步。

（3）从 P_t 中随机选择两个组织 org_{p1} 和 org_{p2}，当组织 org_{p1} 或 org_{p2} 中有一组织所含对象个数为 1 时，则执行合并算子；否则从增减、交换和合并算子中随机选择一个算子，对 org_{p1} 和 org_{p2} 进行相应的操作，产生子代组织 org_{c1} 和 org_{c2}，然后计算每个组织的适应度。

（4）从父代和子代组织中，选择出适应度较高的组织加入到下一代。然后转到第 2 步。

（5）如果进化代数 t 达到了设定的进化代数，则算法结束，从关联规则中输入满足支持度要求的规则集；否则，进化代数 t 的值加 1，转到第 2 步。

通过以上算法，可以得到满足最小支持并要求的规则集，然后，再根据最小置信度的要求，从所得的关联规则集中选择出满足要求的规则集。

24.3 基于组织层次进化的关联规则挖掘

在关联规则算法中，如何快速发现所有频繁项目集是关联规则挖掘过程中最关键的一步。前面提出的基于组织进化的关联规则算法中的进化算子和组织选择算子不能达到这个目的，需要定义一个新的算子。

24.3.1 聚合算子

如果组织 org_{p1} 的相同属性集合 $samp_{org_{p1}}$ 和 org_{p2} 相同属性集合 $samp_{org_{p2}}$ 是相等的，即 $samp_{org_{p1}} = samp_{org_{p2}}$，将组织 org_{p1} 和组织 org_{p2} 合并为一个组织 org_c。这样通过聚合算子可将种群中具有相等的相同属性集合的不同组织聚集在一起，形成一个更大的组织。同时，可以用相同属性集合表来加快组织之间相同属性集合的比较。该表的字段属性如下：序列号、属性名、属性值、记录个数。其中，记录个数的含义是支持该相同属性集合性质的数据个数，它的值是可变的，它最终的值是与规则的支持度相关的，另外，每个组织加了一个标记相同属性集合的标志位，标志位的值即为相同属性集合表的序列号字段的值，如果该组织不存在相同属性集合，则该组织标志位的值为 0。这样，只比较两个组织的标志位，就可以知道两个组织的相同属性集合是否相等。

24.3.2 进化种群 p_e 和最优种群 p_b

进化种群 p_e 是指每条数据对象的数目，在算法初始化时将它们以自由态形式加入。每一代进化结束后，具有相同属性集合的组织数目为最优种群 p_b。种群 p_b 中的组织在进化一定代数后，把其中对象个数少于一定数量的组织解散，其对象以自由态组织形式再加入进化种群 p_e。因为种群 p_e 只包含自由态组织，所以种群 p_e 在进化时只执行合并算子。但对于种群 p_b 在进化时则只执行聚合算子。这样，进化种群 p_e 和最优种群 p_b 中的组织是交替运行同时进行的。另外，还采用组织选择算子从父代组织和子代组织中选择出适应度最高的组织加入下一代进化。

24.3.3 算法步骤

（1）定义 4 个种群：进化种群 p_e、最优种群 p_b、进化暂存种群 p_{et} 和最优暂存种群 p_{bt}；把每一条原始数据以自由态组织的形式加入种群 p_e 中，把种群 p_b、$p_{e\,t}$ 和 p_{bt} 置为空集；令种群 p_e 的进化代数 T_e=0，种群 p_b 的进化代数 T_b=0。

（2）如果当前进化代数 $T_e \leqslant t$，则转为步骤（3）；否则，转步骤（11）。

（3）如果种群 p_{et} 中的组织个数≥1，则把种群 p_{et} 中的组织转入种群 p_e 中。

（4）如果在当前进化代数中，种群 p_e 中未进化的组织个数＞1，则转到步骤（5）；否则转到步骤（6）。

（5）从种群 p_e 中随机选择两个组织 org_{p1} 和 org_{p2}，执行合并算子，产生 org_c；计算组织 org_c

（5）如果进化代数 t 达到了设定的进化代数，则算法结束，从关联规则中输入满足支持度要求的规则集；否则，进化代数 t 的值加 1，转到第 2 步。

通过以上算法，可以得到满足最小支持并要求的规则集，然后，再根据最小置信度的要求，从所得的关联规则集中选择出满足要求的规则集。

24.3 基于组织层次进化的关联规则挖掘

在关联规则算法中，如何快速发现所有频繁项目集是关联规则挖掘过程中最关键的一步。前面提出的基于组织进化的关联规则算法中的进化算子和组织选择算子不能达到这个目的，需要定义一个新的算子。

24.3.1 聚合算子

如果组织 org_{p1} 的相同属性集合 $samp_{org_{p1}}$ 和 org_{p2} 相同属性集合 $samp_{org_{p2}}$ 是相等的，即 $samp_{org_{p1}} = samp_{org_{p2}}$，将组织 org_{p1} 和组织 org_{p2} 合并为一个组织 org_c。这样通过聚合算子可将种群中具有相等的相同属性集合的不同组织聚集在一起，形成一个更大的组织。同时，可以用相同属性集合表来加快组织之间相同属性集合的比较。该表的字段属性如下：序列号、属性名、属性值、记录个数。其中，记录个数的含义是支持该相同属性集合性质的数据个数，它的值是可变的，它最终的值是与规则的支持度相关的，另外，每个组织加了一个标记相同属性集合的标志位，标志位的值即为相同属性集合表的序列号字段的值，如果该组织不存在相同属性集合，则该组织标志位的值为 0。这样，只比较两个组织的标志位，就可以知道两个组织的相同属性集合是否相等。

24.3.2 进化种群 p_e 和最优种群 p_b

进化种群 p_e 是指每条数据对象的数目，在算法初始化时将它们以自由态形式加入。每一代进化结束后，具有相同属性集合的组织数目为最优种群 p_b。种群 p_b 中的组织在进化一定代数后，把其中对象个数少于一定数量的组织解散，其对象以自由态组织形式再加入进化种群 p_e。因为种群 p_e 只包含自由态组织，所以种群 p_e 在进化时只执行合并算子。但对于种群 p_b 在进化时则只执行聚合算子。这样，进化种群 p_e 和最优种群 p_b 中的组织是交替运行同时进行的。另外，还采用组织选择算子从父代组织和子代组织中选择出适应度最高的组织加入下一代进化。

24.3.3 算法步骤

（1）定义 4 个种群：进化种群 p_e、最优种群 p_b、进化暂存种群 p_{et} 和最优暂存种群 p_{bt}；把每一条原始数据以自由态组织的形式加入种群 p_e 中，把种群 p_b、$p_{e\,t}$ 和 p_{bt} 置为空集；令种群 p_e 的进化代数 T_e=0，种群 p_b 的进化代数 T_b=0。

（2）如果当前进化代数 $T_e \leqslant t$，则转为步骤（3）；否则，转步骤（11）。

（3）如果种群 p_{et} 中的组织个数≥1，则把种群 p_{et} 中的组织转入种群 p_e 中。

（4）如果在当前进化代数中，种群 p_e 中未进化的组织个数＞1，则转到步骤（5）；否则转到步骤（6）。

（5）从种群 p_e 中随机选择两个组织 org_{p1} 和 org_{p2}，执行合并算子，产生 org_c；计算组织 org_c

的相同属性集合，如果组织 org_c 有相同属性集合，则把该组织转入种群 p_{be} 中；如果组织 org_c 没有相同属性，则把组织 org_{p1} 和 org_{p2} 标记为已进化，同时删除组织 org_c，转到步骤（4）。

（6）如果种群 p_{bt} 中的组织个数≥1，则把种群 p_{bt} 中的组织加入种群 p_b 中，然后转到步骤（7）；否则，令 $T_e=T_e+1$，转到步骤（2）。

（7）如果当前进化代数 $T_b \leqslant M$，则转到步骤（8）；否则，转到步骤（10）。

（8）如果在当前进化代数中，种群 p_b 中未进化的组织个数 > 1，则转到步骤（9）；否则转到步骤（10）。

（9）从种群 p_b 中随机选择两个组织 org_{p1} 和 org_{p2}，执行合并算子。

（10）如果 $T_b \geqslant M$，则统计种群 p_b 中每个组织的对象个数，把对象个数小于 N 的组织解散，其对象以自由态组织形式转入种群 p_{et} 中，令 $T_b=0$，$T_e=T_e+1$，转到步骤（2）；否则，令 $T_b=T_b+1$，转到步骤（7）。

（11）如果算法满足终止条件，则把种群 p_b 中支持度满足要求的相同属性集合作为关联规则输出；否则，令 $T_e=T_e+1$，转到步骤（2）。

24.4 多维关联规则挖掘

通过对关联规则挖掘的深入研究，可以发现在应用中需要一种能兼顾适应度和支持度条件，同时挖掘出多个关联规则的快速算法。比较各种算法的优缺点，发现多克隆算法符合这一条件，此算法收敛速率快，具有并行性和记忆功能，并且不能导致种群多样性的减弱，具有很强的全局及局部搜索能力。

24.4.1 染色体的编码

多克隆算法建立在编码的基础之上，合适的编码方法会提高后续工作的效率。在此，采用十进制编码。

在关联规则的挖掘中，通过对数据进行概化和归纳，可能会删除一些对数据挖掘没有太大意义的属性列，但保留多个属性列。

多维关联规则挖掘所得到的一般是由各个属性的合取式组成的形如

$$A_1 \wedge A_2 \wedge \cdots \wedge A_n => B_1 \wedge B_2 \wedge \cdots \wedge B_m$$

的规则，可以用这样一个大代码表示：

（1）每个属性对应一个较小的编码段；

（2）这些较小的代码段以同一顺序排列成大的代码段。

在实际操作过程中，可以采用实值编码，假设有一个由 age、income、occupation、iterm_bought 组成的事实表（其中 age 属性有 6 个值，income 有 10 个值、occupation 有 30 个值、iterm_bought 有 25 个值），其编码的范围为 0 0 0 0～6 10 39 25，其中 0 表示这个属性未被选择。

24.4.2 亲和度函数的构造

亲和度函数 f 是评价抗体与抗原联系的量化反映，它的选取对于克隆算法具有非常重要的作用。

在关联规则挖掘中，支持度是对关联规则重要性的衡量，它说明了关联规则在所有事物中的

代表性，它的大小反映了关联规则在实际应用中普遍性的大小。置信度反映了由相关条件结论的正确率，如果置信度达不到一定的阈值，那么这个关联规则就没有意义，所以，选用支持度作为筛选条件，以置信度作为亲和度函数，表示为：$f=C$，其中，C 为置信度。

24.4.3 算法步骤

（1）随机产生每一属性值，以概率 α_i 取 0 选取此属性值，以概率（$1-\alpha_i$）选择其他属性值，其范围为从 1 到此属性值个数间随机选择的一个整数。当某一属性对应的选取概率 $\alpha_i=0$ 时，此属性必存在于所挖掘的关联规则之中，若 α_i 不为 0，则其对应的属性不一定存在于所挖掘的关联规则之中。所以，如果要挖掘出包含特定属性的关联规则时，由应将此属性的选取概率 α_i 取 0，其余属性的概率 α_i 一般取 0.2～0.5。循环选取 n 个初始抗体，这些抗体中各个属性的顺序相同，且应保持每个抗体满足支持度阈值条件。由此形成最初的抗体种群 $\overline{A}(k)$。

（2）计算出每一抗体的 q_i，对抗体种群进化克隆操作 T_c^C，克隆后，种群变为 $\overline{A}'(k)=\{\overline{A}_1'(k),\overline{A}_2'(k),\cdots,\overline{A}_n'(k)\}$。

（3）对目前种群 $\overline{A}'(k)$ 进行克隆变异操作 $\overline{A}''(k)=T_m^c(\overline{A}'(k))$，以概率 P_m^c 从 $\overline{A}'(k)$ 中抽取抗体，对一个或多个属性进行实值变异，使其以一定概率随机变为其他属性值。删除此种群不满足支持度条件的抗体。

（4）对目前种群 $\overline{A}''(k)$ 进行克隆交叉操作 $\overline{A}'''(k)=T_{cr}^c(\overline{A}''(k))$，交叉时使用离散重组法则，删除此种群中不满足支持度条件的抗体。

（5）对目前种群 $\overline{A}'''(k)$ 进行克隆选择操作 $\overline{A}(k+1)=T_s^c(\overline{A}'''(k))$，若得到的某个抗体同时满足最小支持度和最小置信度条件，则输出此抗体，并把此抗体还原为原始属性值，再保留到种群之中。如果迭代次数满足终止条件，则算法结束；否则，把此种群作为下一代计算的初始抗体种群，转到步骤（2）。

24.5 关联规则扩展

24.5.1 多层次关联规则

根据规则中数据涉及的层次关系，可以分为单层次和多层次的关联规则。在单层的关联规则中，没有考虑具体的数据项具有多个不同层次的关系；而多层次的关联规则对数据项的层次关系进行了充分考虑，以在更高的层次发现强关联规则。由于数据存在一定的稀疏性，在低层或原始层的数据项之间很难找到强关联规则，而在较高的概念层发现的强关联规则可能更有意义。因此，多层次关联规则挖掘可以在不同抽象层次上发现更有意义的规则。事务集合中的数据项之间存在一定的概念层次，例如，图 24.1 所示为食品的概念分层示意图，规则“蒙牛酸奶⇒ 黄面包”可能不满足最小支持度要求，可以将黄面包沿概念层次往上提升合并到面包这个层次，规则“蒙牛酸奶⇒ 面包”可能就是强关联规则。

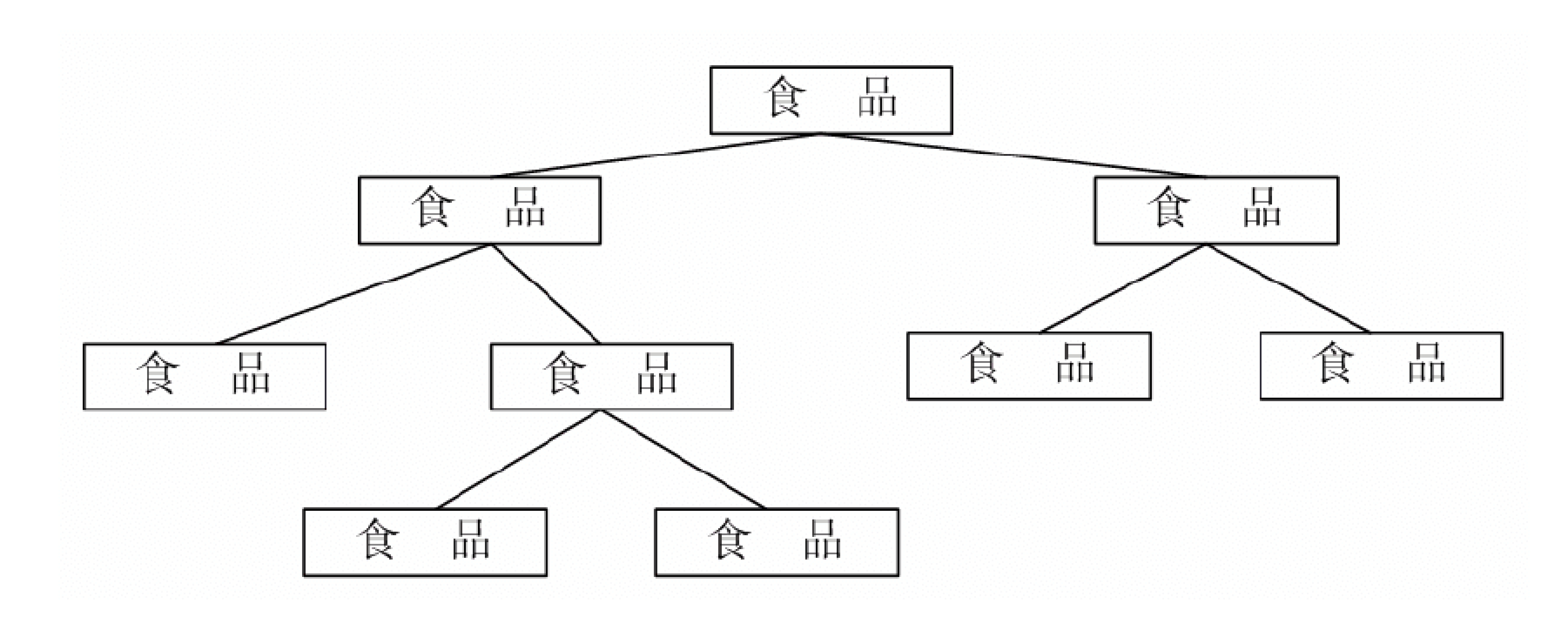


图 24.1　食品的简单概念分层图

多层次关联规则分析算法还是基于传统的经典算法，只是在支持度的设置上做了调整，通常用两种途径。一种是直接对事务数据库应用单层次关联规则挖掘算法，算法将在多层次的数据项中采用相同的支持度。这种方法会出现两种较为极端的结果：支持度太高会丢失低层次有意义的关联规则，支持度太低会产生太多高层次的无意义的关联规则。另一种方法是根据自上而下的思想，先找到高层次的“强”关联规则，再发现低层次的“弱”关联规则，算法需要采用随着层次的降低支持度递减的策略。

24.5.2　多维度关联规则

根据规则中数据项涉及的维度，可以分为单维度和多维度的关联规则。单维度关联规则只涉及数据的一个维，比如“啤酒 ⇒ 尿布”这条规则只涉及用户购买的物品；多维度关联规则涉及数据的多个维度，比如“年龄 = 19” and “职业 = ‘学生’” ⇒ “会买手机”，这条规则就涉及了 3 个维度的信息，是 3 个维度上的关联规则。对于多维数据库或数据仓库而言，实际挖掘的都是多维度关联规则。

多维度关联规则的挖掘关键在于搜索频繁项 k-维词集合，比如{“年龄”，“职业”，“是否会买手机”}是一个 3-维词集合。搜索前，需要对不同类型的属性进行处理。对于数值属性，可以用预定义的概念层次或其他静态方法进行离散化，也可以根据数据分布对数值属性分箱来达到动态的离散化，还可以利用数据点之间的距离实现动态的离散化。比如“年龄”数值属性离散化，划分到哪个年龄之间，属于哪个年龄阶段等。在多维数据库中搜索频繁项 k-维词集合，需要 k 或 k+1 次表扫描，使用数据立方体可以实现更快的搜索。

24.5.3　定量关联规则

与布尔型关联规则处理离散化的属性不同，定量关联规则挖掘是从包含连续属性的数据集中挖掘关联规则。为了得到定量关联规则，需要对连续属性进行离散化，从而将问题转化为布尔关联规则挖掘。定量关联规则是多维关联规则的一种，可以称为带数值的关联规则。因此需要对其中的数值属性离散化，将其转化为布尔型关联规则。根据实际数据的特点，将每个属性值映射为一个布尔型属性，可以采用等宽分箱（每个箱的区间长度相同）、等深分箱（每个箱赋予大致相同个数的元组）、基于同质的分箱（箱的大小应使得每个箱的元组一致分布）等分箱技术，来实现对数值属性的离散化。

24.5.4 基于约束的关联规则

基于约束的挖掘方式以用户为驱动，此时用户应具有较好的规则判断能力，知道什么形式的规则对他们有价值。一种更有效产生关联规则的方法是让用户说明他们的直觉或期望作为限制搜索空间的约束条件。这些约束包括如下几种。

（1）知识类型约束：指定要挖掘的知识类型，如关联规则或相关规则。

（2）数据约束：指定任务相关的数据集。

（3）维层约束：指定所用的数据，或概念分层结构的层次。

（4）兴趣度约束：指定规则兴趣度统计度量阈值，如支持度、置信度或其他评估度量。

（5）规则约束：指定要挖掘的规则形式。这种规则可以用规则模板表示。

24.6 例题

例 4.66 考查表 24.1 所列的模拟购物事务数据库，每一条数据由事务的发生时间、购买客户的 ID 以及购买的项目 ID 组成。试求其中的关联规则。

表 24.1 事务数据库

时 间	客 户 ID	项 目 ID	时 间	客 户 ID	项 目 ID
07－06－10	002	10,20	07－06－25	004	30
				003	30,50,70
				001	30
07－06－12	005	90	07－06－30	001	90
				004	40,70
07－06－15	002	30	07－07－25	004	90
07－06－20	002	40,60,70			

解：

对于这类问题可以用 AprioriAll 算法解决。此算法是 Apriori 算法的扩展，它首先将时间作为标识的事务数据库转换为以顾客作为标识的序列数据库 SD，每一顾客唯一对应一个项目集表示的序列模式，然后利用 Apriori 算法对所生成的序列数据库求其关联规则。

根据以上原理，可编程计算如下。

```
>>x={2 {'10','20'};5 {'90'};2 {'30'};2 {'40','60','70'};4 {'30'};3 {'30','50',
     '70'};1 {'30'};1 {'90'};4 {'40','70'};4 {'90'}};
sup_min=0.2; conf_min=0.5;
>> [y1,y2]=aprioriall(x,sup_min,0.5);
>> y1='5→1  conf=1'                                    %关联规则
 '4→1  conf=0.66667'
 '1→5  conf=0.66667'
 '1→4  conf=0.66667'
```

```
>>y2='30'  '40'  '70'  '90'  {1×2 cell}  {1×2 cell}    %规则中各数字的意义
    y2{5}={'40'  '70'}
    y2{6}={'30'  '70'}
```

例 4.67　在进行股票投资中，投资者希望看到的是某几只股票间的联动关系以及可能性。较常见的有 2、3、4 支股票间的联动上涨或下跌，在这种情况下，可将其转换为关联规则："股票 A 在 T_a 时间上涨且股票 B 在 T_b 时间上涨，则股票 C 在 T_c 时间上涨，支持度是 X%，置信度是 Y%"。下面对表 24.2 中的数据进行关联分析。表中的 1 表示"涨"，0 表示"跌"，T_i 为某一连续时间段，I_i 为表示某一股票。最小支持度为 0.5，最小可信度为 0.7。

表 24.2　某时间段内股票涨跌情况

时　间 / 股　票	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
I1	1	0	1	1	1	0	1	0
I2	0	1	1	0	1	0	1	0
I3	1	0	1	1	1	0	1	1
I4	0	1	1	0	1	0	1	1

解：

在进行这类关联规则挖掘中，最常见的求 3 支股票间的关系，所以最大项为 3－频繁项，挖掘规则为：A(1)+B(2)->C(4)，即 B 股票与 A 股票相差 1 天，2－频繁项 A、B 与 C 股票相差 4 天时。据此，可以编程计算如下。

```
>> x=[1 0 1 1 1 0 1 0;0 1 1 0 1 0 1 0;1 0 1 1 1 0 1 1;0 1 1 0 1 0 1 1];
sup=0.5;conf_min=0.7;
>> y=bitsearch(x,sup,conf_min);
    y=1.0000    2.0000    3.0000    0.5774    1.0000
      3.0000    2.0000    1.0000    0.6325    1.0000
```

挖掘出两条规则，即：

（1）第 1 天 I1 股票上升，第 2 天 I2 股票上升，第 4 天 I3 股票上升，支持度为 0.5774，置信度为 1.0。

（2）第 1 天 I3 股票上升，第 2 天 I2 股票上升，第 4 天 I1 股票上升，支持度为 0.6325，置信度为 1.0。

例 4.68　不同于 Apriori 等算法的"产生—测试"模型，深度优先法不产生频繁项集，而是采用模式增长的方式产生关联规则。深度优先法典型代表是 FP-growth 算法，该算法使用一种 FP 树的紧凑数据结构数据，经过一次扫描后，将数据库中的事务压缩到一棵频繁模式树中，采用分而治之的策略，对频繁模式树进行处理，其主要通过减少 I/O 次数来提高效率。但 FP-growth 算法仍存在着一些影响其挖掘效率的重要因素：首先在挖掘频繁模式时，需要递归的生成条件模式基和条件模式树；其次，在频繁模式挖掘过程中，需要动态生成、销毁大量的条件模式树，这将消耗大量的时间和空间。

为了改进 FP-tree 数据结构，可以采用单向链表的结构来储存节点信息，该链表按照项目的

支持度降序来排列。通过遍历各个节点的单向链可以得到相应的频繁模式树，该算法也只需扫描两次事务数据库，同时避免产生大量的条件模式树，从而节省了时间和空间，提高了效率。

利用此法对表 24.3 中的数据进行关联规则分析，其最小支持数为 2。

表 24.3　事务数据库 D

TID	Iterm
1	a,c,d
2	b,c,e
3	a,b,c,e
4	b,e

解：

表 24.3 事务数据集 TDB 中的每条事务的项目集实际上是顺序扫描时逐条产生的，每产生一条项目集，便作为路径递归分割插入到链表中。例如插入<b,c,e>，先将 b 看作项目头节点，其后插入< c,e>，再把 c 看作项目头节点，其后插入<e>，因为原先 c 后已经有<a>了，此时要插入<e>，因为在项头表中 e 在 a 的前面，所以将<e>插在<a>的前面。对于本例最终的邻接表为图 24.2 所示的结构。从邻接表可以方便地得出各项集频繁项。对于每一个以项头表中的一个节点为头节点的各单链表求频繁项目集时，是必须包含项头表中的这个节点的各支持度不小于 2 的所有节点的排列组合，其组合后的支持度为其中最小的支持度。例如对于节点 b 而言，其组合< bce >、< bc >、< be >均为频繁项。对于各频繁项再进行支持度的判断，便可以最终求出最大频繁项。

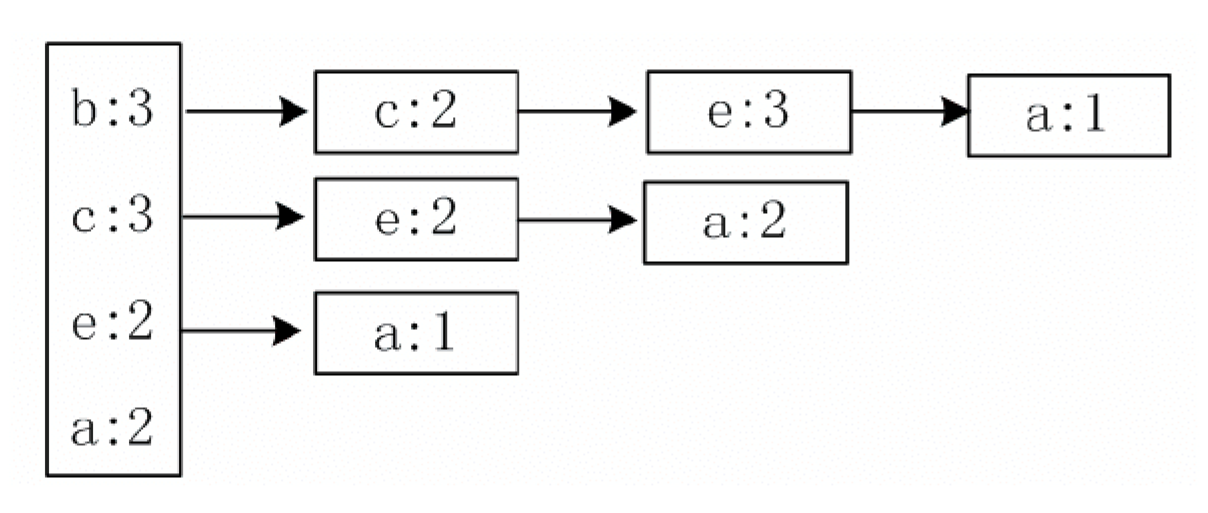


图 24.4　邻接表结构

根据此算法的原理，便可以编程进行计算。

```
>> x={{'a','c','d'};{'b','c','e'};{'a','b','c','e'};{'b','e'}};
>>sup=2;
>> y=nearjoin(x,sup);
 y = 'c'    'b'    'e'    %最大频繁项集
```

例 4.69　目前，关联规则挖掘算法一般是指 Apriori 算法或其改进算法，其挖掘过程主要包含两个阶段：第一阶段必须先从数据集合中找出所有支持度大于用户最小支持度的项集的频繁项目集；第二阶段由这些频繁项目集构造可信度大于用户最小可信度的关联规则。它实际上是一个全局搜索过程。遗传算法是一种全局优化算法，它可以有效地避免搜索过程的局部最优解问题，因此，将遗传算法用于关联规则的发现和提取有利于找到有价值的规则。

试利用遗传算法对表 24.4 的数据进行关联规则挖掘，其中最小支持度数为 2。

表 24.4　事务数据库 D

TID	Iterm
1	I1，I2，I5
2	I2，I4
3	I2，I3
4	I1，I2，I4
5	I1，I3
6	I2，I3
7	I1，I3
8	I1，I2，I3，I5
9	I1，I2，I3

解：

遗传算法中各参应根据实际情况设置。对于本例，参数设置如下：

编码方式：采用二进制编码，1代表该项出现，0为不出现。例如编码[01100] 表示该项集为I2、I5。

编码长度：因为事务数据库中的项数为5，根据出现顺序其排列为I1、I2、I5、I4、I3，所以设定编码的长度为5。

交叉率P_c：0.85

变异率P_m：0.01

种群大小：50

迭代次数：20

适应度函数：$f(x)=\begin{cases}-\text{sum}(x) & \text{sup} \geqslant \text{sup_min} \\ \text{size}(M,2) & \text{sup} < \text{sup_min}\end{cases}$，其中M为数据库的项数，在这里为5。

根据以上参数，首先编写适应度函数如下。

```
function
y=rule_ga(x)data={{'I1','I2','I5'};{'I2','I4'};{'I2','I3'};{'I1','I2','I4'};
                    {'I1','I3'};{'I2','I3'};{'I1','I3'};{'I1','I2','I3','I5'};
                    {'I1','I2','I3'}};
sup=2;num=size(data,1);
[iterm,supcount]=myiterm(data);          %求项数及对应的支持度数
a=length(iterm);y=zeros(num,a);
for i=1:num
    for j=1:a
       [y1,y2]=mycompare1(iterm,data{i,:});
       if ~isempty(y1)
          y(i,y2)=1;                       %表示某项数在事务中是否出现的矩阵
```

```
      end
   end
end
a1=dot(x,x);b=y*x';a2=find(b==a1);      %计算支持度
if length(a2)>=sup; y=-length(find(x));else; y=a;end
```

据此，便可利用优化工具OPTIMTOOL（或GATOOL）GUI计算。在solve框中选择ga-Genetic Algorithm，在fitness Function框中输入@ rule_ga，在Number of variables框中输入5，在Population type框中选择bit string，在Population size选择Specify并输入“50”，在Crossover fraction选择Specify并输入0.85，在Mutation框中，rate选择Specify并输入“0.01”，其余选择默认值，最后运行ga。

经过多次运算，可以得到两个最长频繁项：I1、I2、I5与I1、I2、I3。

例 4.70 遗传算法在关联规则挖掘中的另外一种应用方式是直接将关联规则作为个体，利用遗传算法对其进行优化。

某商场为了更好地营销某品牌商品，最近做了问卷调查，主要了解该商品的价格（分高、中和低）、质量（分优、一般和差）、售后服务（分好和差）和满意程度（分非常满意、满意、接受和不接受）等情况。试用遗传算法对问卷情况进行分析。

解：

根据情况，将规则作为个体进行优化，以挖掘出符合条件的规则集。为此将规则进行编码。一条规则用4个实整数表示，其中价格用1、2和3分别代表高、中和低，质量用1、2和3分别代表优、一般和差，用1和2分别代表售后服务的好和差，用1、2、3和4分别代表满足程度的4种情况。因此规则1324表示如果该商品价格高、质量差、服务差，则顾客不会接受。

因为本例只是作为一个实例说明遗传算法在关联规则挖掘中的应用，所以模拟产生问卷，所随机产生的答案即规则不一定符合实际情况。

各参数设置如下：

编码方式：实整数

交叉率P_c：0.85

变异率P_m：0.05

种群大小：30

迭代次数：100

最小支持度：0.1

最小置信度：0.7

适应度函数：$f(x)=a\times Sup+b\times Confid+c\times Cover$，其中Sup为规则的支持度、Confid为规则的置信度、Cover为规则的覆盖度，a、b和c为三个[0、1]间的常数。

据此，便可以编程进行计算（也可以利用遗传算法的GUI计算）。

```
>>rand('state',1);                              %保持随机数一致
x=[myrandn(800,1,3,'i') myrandn(800,1,3,'i') myrandn(800,1,2,'i')
myrandn(800,1,4,'i')];                          %数据库
```

```
x=repeat_set(x);                              %去掉数据库中矛盾的规则
m=30;pc=0.85;pm=0.05;t=100;vara_bound=[3 3 2 4];sup=0.1;conf=0.7;a=1;b=1;c=1;
>> pop_best=asso_rules_ga(x,sup,conf,m,t,pc,pm,vara_bound,a,b,c);
```

计算所得规则如下：

规则1

```
rule1: [2 2 1 2]
      confid: 1               %置信度
      lift: 0.0034            %提升度（兴趣度）
     cover: 0.3741            %覆盖度
       fit: 111.3741          %适应度
       sup: 110               %支持度数
```

规则2

```
      rule: [2 2 2 4]
    confid: 1
      lift: 0.0048
     cover: 0.5646
       fit: 119.5646
       sup: 118
```

例 4.71　在实际中会经常遇到类似例 4.70 的多值属性关联规则的挖掘。多值属性根据属性性质可分为数值和类别属性，前者可以转化成类别属性。多值属性关联规则的挖掘更为复杂。在挖掘之前首先要对多值属性（数值属性）进行合理的划分，如何合理、有效地划分属性区间，使其能够真实地反映此属性中数据在定义域中的实际分布则是挖掘多值属性关联规则的关键问题。

在完成属性划分后，有两种方法可以实现对多值属性关联规则的挖掘。多数文献主张将多值关联规则问题转化为布尔型关联规则问题，这需要将多值数据转化为布尔型数据，但由此引出了一系列新的问题，比如布尔型关联规则中定义的支持度和关联度概念在多值属性关联规则中是否适用；将多值数据转化为布尔型数据后增加了大量的存储空间等；另一种方法则借助其他数学工具实现。

试利用第一种方法对表 24.5 所示的天气数据进行关联规则挖掘。

例 24.5　天气样本数据

属　性	Outlook	Temperature	Humidity	Windy	类　别
1	Overcast	Hot	High	Not	N
2	Overcast	Hot	High	Very	N
3	Overcast	Hot	High	Medium	N
4	Sunny	Hot	High	Not	P
5	Sunny	Hot	High	Medium	P
6	Rain	Mild	High	Not	N

续表

属　性	Outlook	Temperature	Humidity	Windy	类　别
7	Rain	Mild	High	Medium	N
8	Rain	Hot	Nornal	Not	P
9	Rain	Cool	Nornal	Medium	N
10	Rain	Hot	Nornal	Very	N
11	Sunny	Cool	Nornal	Very	P
12	Sunny	Cool	Nornal	Medium	P
13	Overcast	Mild	High	Not	N
14	Overcast	Mild	High	Medium	N
15	Overcast	Cool	Nornal	Not	P
16	Overcast	Cool	Nornal	Medium	P
17	Rain	Mild	Nornal	Not	N
18	Rain	Mild	Nornal	Medium	N
19	Overcast	Mild	Nornal	Medium	P
20	Overcast	Mild	Nornal	Very	P
21	Sunny	Mild	High	Very	P
22	Sunny	Mild	High	Medium	P
23	Sunny	Hot	Nornal	Not	P
24	Rain	Mild	High	Very	N

解：

首先对表中的数据进行离散化，即对属性进行分类，每个属性有L种取值，则编码长度L。例如“Outlook”有三种状态，则第一种状态用100、第二种状态有010，第三种状态用001表示，以此类推，这样就可以将数据转化为0、1矩阵。然后对此矩阵进行运算就可以求出各种状态的支持度，进而求出规则。

根据此原理，可编程计算如下。

```
>> x={'overcast' 'hot' 'high' 'not' 'N';'overcast' 'hot' 'high' 'very' 'N';
      'overcast' 'hot' 'high' 'medium' 'N';'sunny' 'hot' 'high' 'not' 'P';
      'sunny' 'hot' 'high' 'medium' 'P';'rain' 'mild' 'high' 'not' 'N';
      'rain' 'mild' 'high' 'medium' 'N';'rain' 'hot' 'normal' 'not' 'P';
      'rain' 'cool' 'normal' 'medium' 'N';'rain' 'hot' 'normal' 'very' 'N';
      'sunny' 'cool' 'normal' 'very' 'P';'sunny' 'cool' 'normal' 'medium' 'P'
      'overcast' 'mild' 'high' 'not' 'N';'overcast' 'mild' 'high' 'medium' 'N';
      'overcast' 'cool' 'normal' 'not' 'P';'overcast' 'cool' 'normal' 'medium' 'P';
      'rain' 'mild' 'normal' 'not' 'N';'rain' 'mild' 'normal' 'medium' 'N';
      'overcast' 'mild' 'normal' 'medium' 'P';'overcast' 'mild' 'normal' 'very' 'P';
      'sunny' 'mild' 'high' 'very' 'P'sunny' 'mild' 'high' 'medium' 'P';
```

续表

属　性	Outlook	Temperature	Humidity	Windy	类　别
7	Rain	Mild	High	Medium	N
8	Rain	Hot	Nornal	Not	P
9	Rain	Cool	Nornal	Medium	N
10	Rain	Hot	Nornal	Very	N
11	Sunny	Cool	Nornal	Very	P
12	Sunny	Cool	Nornal	Medium	P
13	Overcast	Mild	High	Not	N
14	Overcast	Mild	High	Medium	N
15	Overcast	Cool	Nornal	Not	P
16	Overcast	Cool	Nornal	Medium	P
17	Rain	Mild	Nornal	Not	N
18	Rain	Mild	Nornal	Medium	N
19	Overcast	Mild	Nornal	Medium	P
20	Overcast	Mild	Nornal	Very	P
21	Sunny	Mild	High	Very	P
22	Sunny	Mild	High	Medium	P
23	Sunny	Hot	Nornal	Not	P
24	Rain	Mild	High	Very	N

解：

首先对表中的数据进行离散化，即对属性进行分类，每个属性有L种取值，则编码长度L。例如“Outlook”有三种状态，则第一种状态用100、第二种状态有010，第三种状态用001表示，以此类推，这样就可以将数据转化为0、1矩阵。然后对此矩阵进行运算就可以求出各种状态的支持度，进而求出规则。

根据此原理，可编程计算如下。

```
>> x={'overcast' 'hot' 'high' 'not' 'N';'overcast' 'hot' 'high' 'very' 'N';
    'overcast' 'hot' 'high' 'medium' 'N';'sunny' 'hot' 'high' 'not' 'P';
    'sunny' 'hot' 'high' 'medium' 'P';'rain' 'mild' 'high' 'not' 'N';
    'rain' 'mild' 'high' 'medium' 'N';'rain' 'hot' 'normal' 'not' 'P';
    'rain' 'cool' 'normal' 'medium' 'N';'rain' 'hot' 'normal' 'very' 'N';
    'sunny' 'cool' 'normal' 'very' 'P';'sunny' 'cool' 'normal' 'medium' 'P'
    'overcast' 'mild' 'high' 'not' 'N';'overcast' 'mild' 'high' 'medium' 'N';
    'overcast' 'cool' 'normal' 'not' 'P';'overcast' 'cool' 'normal' 'medium' 'P';
    'rain' 'mild' 'normal' 'not' 'N';'rain' 'mild' 'normal' 'medium' 'N';
    'overcast' 'mild' 'normal' 'medium' 'P';'overcast' 'mild' 'normal' 'very' 'P';
    'sunny' 'mild' 'high' 'very' 'P'sunny' 'mild' 'high' 'medium' 'P';
```

```
    'sunny' 'hot' 'normal' 'not' 'P';'rain' 'mild' 'high' 'very' 'N'};
>>sup=0.2;conf=0.6;                              %最小支持度和最小置信度
>>[rule,L_max,L_iter]=mulit_rule(x,sup,conf);    %求多重属性关联规则的函数
```

其中rule为符合条件的规则，L_max为最大频繁项长度，L_iter为各项，L_iter1第1行为属性，第二行为对应属性的取值。

例：rule{1}

```
  '1→2  5;  支持度=5;  置信度=0.625'
  '1  2→5;  支持度=5;  置信度=1'
  '1  5→2;  支持度=5;  置信度=0.71429'
  '2  5→1;  支持度=5;  置信度=0.71429'
  L_max=3;
  L_iter{1}= 'rain'   'mild'   'N'               %最大频繁项中的一个
  >> L_iter1{1}=1     2     5                    %属性
                2     3     1                    %属性对应的取值
```

例 4.72　聚类和关联规则是数据挖掘研究的重要内容。聚类和关联规则挖掘的目的在于通过分析大量数据，从中找出人们未知的却又具有潜在使用价值的规则。

在对大型数据库进行关联规则挖掘时，为了减少运行时间，可以先通过聚类方法分析数据集，而后通过关联分析来对每一个聚类簇（具有相似特性的数据集）的一些特征进行分析，找出每一类所具有的共同特征或规则。

为了更好地提高教学效果，对以往学生的“计算机基础”课程成绩进行分析，找出影响该课程成绩的关键因素，进而采取适当的措施以提高课程的教学质量。

因篇幅关系，在表 24.6 中只列出模拟的 200 位学生中 10 位学生的“计算机基础”综合成绩。该课程由基础知识、基本操作、office 操作和网络使用 4 个模块组成，而基础知识又有单选题、判断题、多选题 3 种题型；基本操作有汉字输入、Windows 操作 2 种题型；Office 操作有 Word 操作、PPT 操作和 Excel 操作 3 种题型；网络使用有信息浏览题、E-mail 操作题 2 种题型。成绩已作规范化处理。

表 24.6　计算机基础课程成绩

模块 / 课程信息	基础知识	基本操作	Office 操作	网络操作
理科	0.9730	0.9748	0.1217	0.6862
理科	0.1892	0.6513	0.8842	0.8936
文科	0.6671	0.2312	0.0943	0.0548
文科	0.5864	0.4035	0.9300	0.3037
艺术	0.6751	0.1220	0.3990	0.0462
理科	0.3610	0.2684	0.0474	0.1955
理科	0.6203	0.2578	0.3424	0.7202

续表

模　块 / 课程信息	基础知识	基本操作	Office 操作	网络操作
艺术	0.8112	0.3317	0.7360	0.7218
体育	0.0193	0.1522	0.7947	0.8778
文科	0.0839	0.3480	0.5449	0.5824

解：

首先利用kmeans方法分别对4个模块及10个课程属性进行分类，然后再对分类后的每一类学生进行关联规则分析，以获得各类学生的特点。

分类时考虑到学生成绩的分布，将其分为优秀、良好、中等、不及格与差5类；而每个题型分为优、良与差3个等级。

```
>> mydata=[myrandn(100,1,[1 4],'i') myrandn(100,4,[0 1],'a')];
                                             %产生模拟数据库
>> [a,b]=kmeans(mydata(:,2:end),5);          %分成5类
>> a1=mydata(find(a==1),1);                  %第1类的学生
>> length(find(a1==1));                      %第1类学生中理科生的数目
 ans =8
>> length(find(a1==2))                       %第1类学生中文科生的数目
 ans =7
>> length(find(a1==3))                       %第1类学生中艺术生的数目
 ans =7
>> length(find(a1==4))                       %第1类学生中体育生的数目
  ans =3
```

可以看出，体育生占第1类的比例较少。对于其他类的学生可以作类似的分析，并且根据支持度及置信度寻找出关联规则。

然后对每个题型作关联规则分析：

```
>> mydata1=[myrandn(100,10,[1 3],'i')];          %模拟产生数据库，其格式如下
>> mydata1(1:6,:)
 ans =[2    1    1    2    1    2    3    1    3    1
       2    1    2    2    3    2    3    2    2    1
       3    1    1    1    2    3    3    1    1    1
       3    1    3    2    2    3    1    2    2    2
       2    2    3    2    1    3    1    2    1    1
       2    1    2    3    2    2    1    2    2    2];
```

同样对这个数据集中的第1类数据进行关联规则分析，它是一个多值属性矩阵，可以利用例

4.52中的函数进行分析：

```
>> a2=mydata1(find(a==1),:);                          %第1类的数据集
>> [rule,L_max,L_iter,L_iter1]=mulit_rule(a2,0.3,0.7); %限于篇幅，结果不再列出
```

例 4.73　经典的 Apriori 方法在挖掘项目数较多的大型数据库会遇到效率较低的问题。可以用多种方法解决这个问题，其中一个方法是对大型数据库先进行分类，然后再对每个类进行关联规则分析。

聚类的方法有多种，主要是利用项的距离及其他特点进行。试利用相应的聚类方式对表 24.7 的数据进行关联规则分析。

表 24.7　事务数据库

事　务	项　目
01	'I1'　'I2'　'I5'
02	'I2'　'I4'
03	'I2'　'I3'
04	'I1'　I2'　'I4'
05	'I1'　'I3'
06	'I2'　'I3'
07	'I1'　'I3'
08	'I1'　'I2'　'I3'　'I5'
09	'I1'　I2'　'I3'

解：

先根据项出现的频率对项目进行排序，然后根据每个事务的第一个项目名称进行分类，然后再对每个类进行关联规则分析。

事实上根据此法分类后，每个类即为一棵Fp-tree树，这样可以利用相应的Fp-tree算法对每个类进行关联规则分析，当然也可以用传统的关联规则挖掘算法进行分析。

要注意的是，无论是利用基于项的距离还是本方法的分类方法都应考虑以下问题。

（1）不能遗漏关联规则。有可能某些项在每个类中都不符合支持度要求，但在整个数据库中符合支持度要求。但在计算最大频繁项时不会出现此类情况。

（2）要注意支持度的计算。某些项在不同类都会出现，或者某些项在某个类中是频繁项，但在另外一个类中并不是频繁项，所以在计算相应频繁项的支持度时要考虑这种情况的处理。

根据以上方法的原理，便可以编程进行相应的计算：

```
>> x={{'I1','I2','I5'};{'I2','I4'};{'I2','I3'};{'I1','I2','I4'};
      {'I1','I3'};{'I2','I3'};{'I1','I3'}{'I1','I2','I3','I5'};
      {'I1','I2','I3'}};
>>sup=0.2;
>> y=cluster_rule(x,sup);                      %计算函数，省略规则置信度的计算
```

```
>> y=iter: {{5x3 cell}  {1x4 cell}  {0x5 cell}} %各个频繁项及对应的支持度数
     iterm: {'I2'  'I1'  'I3'  'I4'  'I5'}       %整个数据库符合支持度数要求的项
      sup: [7 6 6 2 2]                           %各个项对应的支持度数
    Liter: {'I2'  'I1'  'I3'  [2]}               %最大频繁项及对应的支持度数
```

例 4.74　在进行关联规则挖掘时，通常会产生成百上千或成千上万条关联规则，要从如此庞大的数量中去发现感兴趣的规则，用户通常需要浏览全部的结果，这显然不是一件容易的事情。此时可以对关联规则进行分类，删除冗余规则，找出可能感兴趣的规则。

试对例4.73所形成的规则进行分类分析。

解：

在进行分类分析时，首先要确定规则间距离的计算方法。规则间距离根据各规则中项集间距离而来。

两个项间的距离可以用下式计算

$$d(I_i,I_j)=1-\frac{s(I_i\cup I_j)}{s(I_i)+s(I_j)-s(I_i\cup I_j)}$$

式中：s为各项的支持度。如果是多个项则是它们的平均距离。

而规则的距离则可以用下式计算

$$d(r_i,r_j)=\alpha\times d(X_1\cup Y_1,X_2\cup Y_2)+\beta\times d(X_1,X_2)+\gamma\times d(Y_1,Y_2)$$

$$\alpha+\beta+\gamma=1$$

式的第一项为两个规则前、后件的并集间的距离，第二项为规则前件间的距离，第三项为规则后件间的距离。α、β和γ为用户自己设定的三个常数，其值越大，说明此部分在距离中的权重越大，所以如果用户关心的是规则前件的相关性，则可设定较大的β值。

根据以上的计算方法，便可以编程对规则进行聚类分析：

```
>> x={{'I1','I2','I5'};{'I2','I4'};{'I2','I3'};{'I1','I2','I4'};
      {'I1','I3'};{'I2','I3'};{'I1','I3'}{'I1','I2','I3','I5'};
      {'I1','I2','I3'}};
>>sup=0.2;conf=0.5;
>> [w1,rule,data,iterm,sup]=myapriori(x,sup,conf); %求规则函数
>> [h1,h2]=claasify_rule(rule,data,sup);           %规则分类函数,见图24.1所示
```

从图中聚类树可明显看出，规则的分类情况，可以视实际情况将零时规则分成不同数目的类。

从函数中还可以得到项集的分类情况，所以此函数还可以用于项集较多时关联规则的挖掘。图24.2即为项集的分类情况。

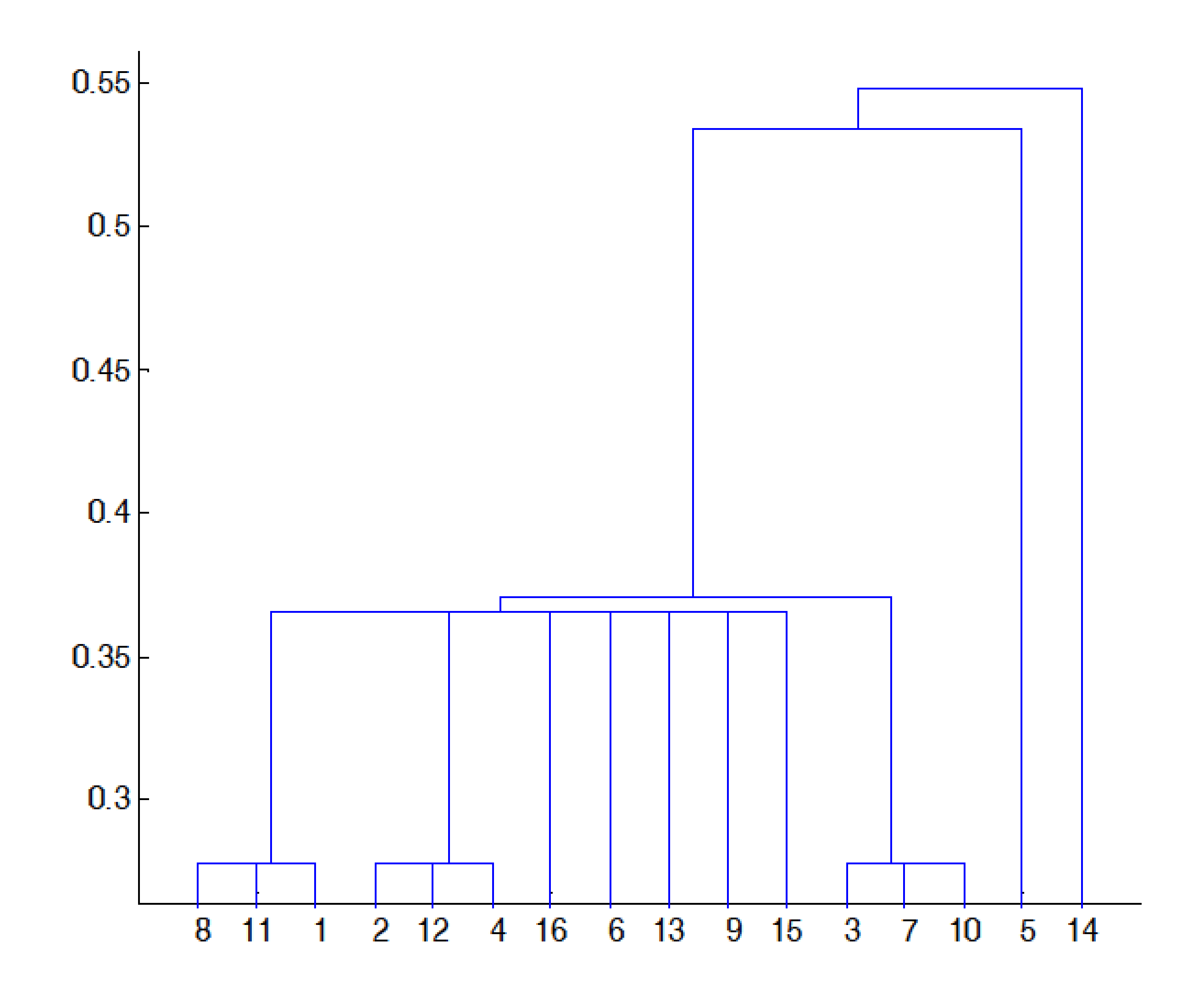


图 24.1　规则的聚类树

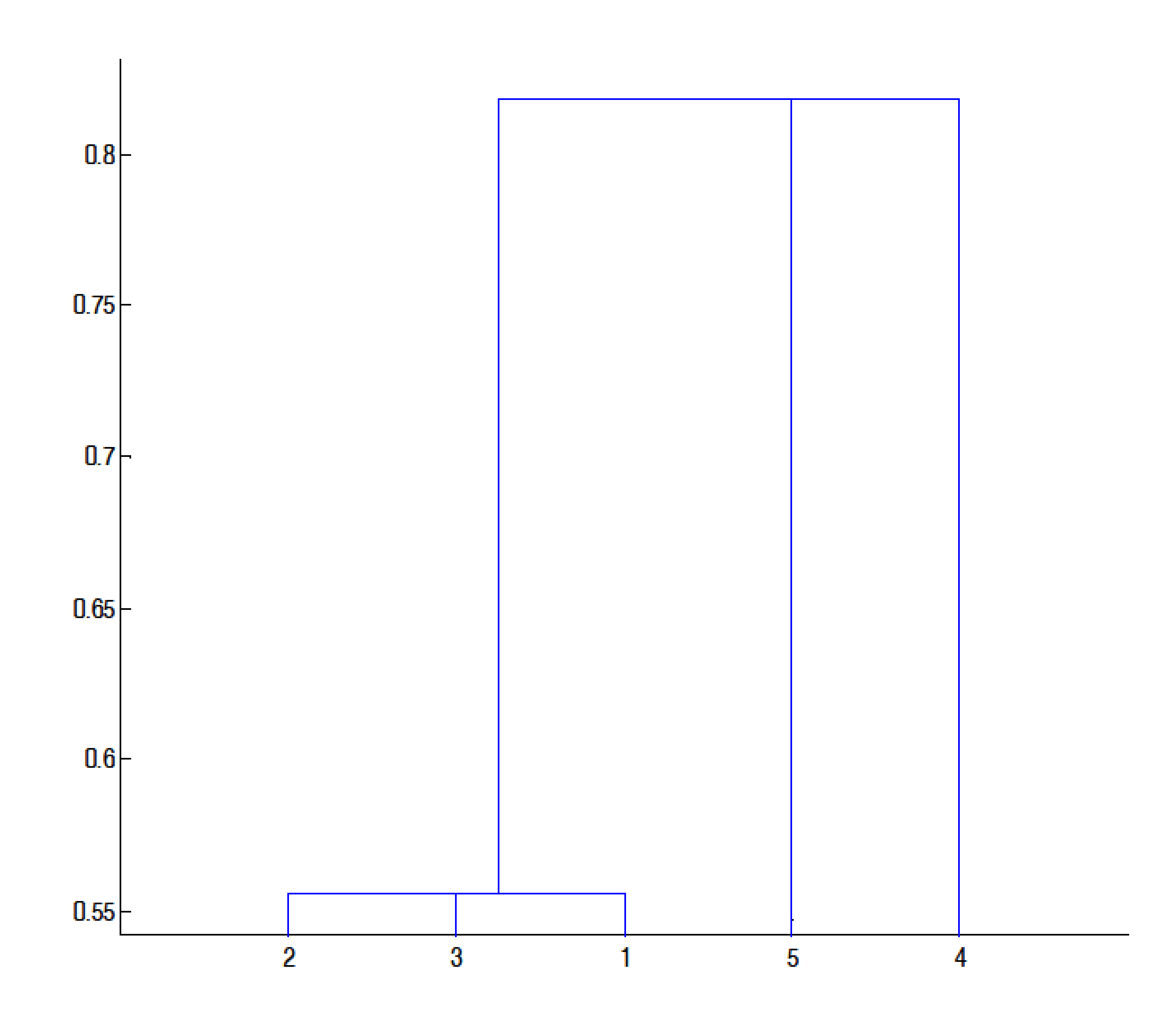


图 24.2　项集的聚类树

例 4.75　关联规则中还有一种多层关联规则。在挖掘多层关联规则时，规则中的项目可以属于同一概念层，也可以属于不同的概念层。

多层关联规则的挖掘过程和传统的关联规则挖掘过程一样：先找出所有频繁项集，再通过频繁项集产生强关联规则，关键在于第一步。可以用多种方法来进行相关的挖掘。

试对表 4.8 的数据进行关联规则挖掘，其中各项（按衣服、外衣、上衣、内衣、裤子、鞋子、皮鞋、拖鞋、帽子的顺序）的最小支持度为 [2/3 1/4 1/6 1/4 1/12 1/3 1/12 1/6 1/6]。

表 24.8　事务数据库

事　务	项　目
01	上衣、拖鞋
02	内衣、帽子
03	内衣、裤子
04	拖鞋、内衣、上衣
05	帽子
06	皮鞋

解：

按照一般的约定，很明显事务数据库的项是不同层次的概念，存在着以下的关系，即子孙（下层）－祖先（上层）的关系，如图24.3所示。

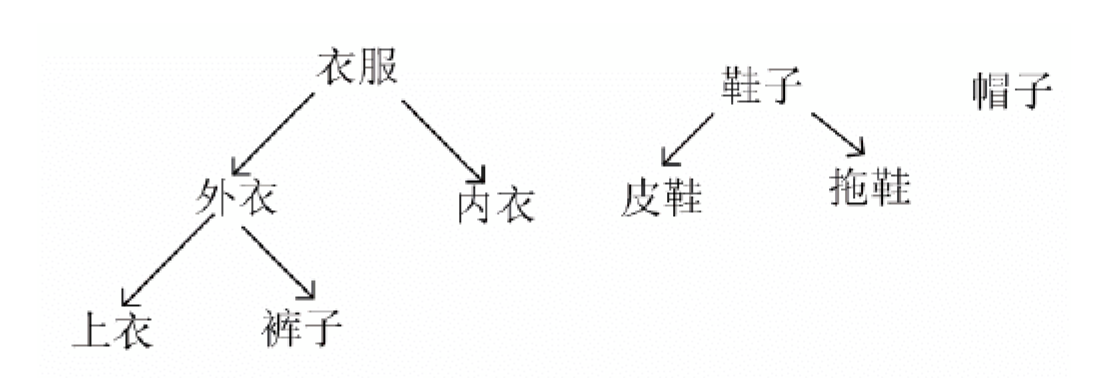


图 24.3　关系图

在进行多层关联规则挖掘时，应注意以下几点。

（1）在对项计数时，子孙项出现表示祖先项也出现。

（2）存在子孙－祖先关系的项不能同时出现。

（3）项的支持度要满足各项最小支持度中的最大值。

据此，便可以编程计算：

```
>> x={{'拖鞋 ' '上衣'};{'帽子' '内衣'};{'内衣' '裤子'};{'拖鞋' '内衣' '上衣'};
      {'帽子'};{'皮鞋'}};
>>mins={{'衣服'  2/3};{'外衣'  1/4};{'上衣'  1/6};{'内衣' 1/4};{'裤子' 1/12};
        {'鞋子' 1/3};{'皮鞋' 1/12};{'拖鞋' 1/6};{'帽子' 1/6}};
>>ia={{'外衣 ' '衣服'};{'上衣' '衣服' '外衣'};{'内衣' '衣服'};{'裤子' '衣服' '外衣'};
      {'皮鞋' '鞋子'}{'拖鞋' '鞋子'}};
>> [L,rule,L_iter,L_d]=rule_Lay(x,mins,ia);  %多层关联规则挖掘函数
>> L{1}= '外衣'    '内衣'          %
         '外衣'    '鞋子'
         '外衣'    '拖鞋 '
         '鞋子'    '上衣'
         '上衣'    '拖鞋 '
```

在应用函数计算时，应注意以下几点。

（1）输入的是支持度，而不是支持度数；

（2）L表示频繁项；

（3）rule是以数字表示的项及相应的支持度；

（4）L_iter表示的是最后出现的项；

（5）L_d表示的是样品项关系的0、1数据矩阵。根据这个矩阵、频繁项及项便可以计算频繁项的支持度，进而求出各规则，限于篇幅在此就不再计算。读者可以根据前面例题中相应的函数或自己编程进行计算。

参 考 文 献

[1] JiaweiHan，MichelineKamber 著. 范明，孟小峰译. 数据挖掘概论与技术（原书第 2 版）. 北京：机械工业出版社，2007.

[2] 廖芹，郝志峰，陈志宏. 数据挖掘与数学建模. 北京：国防工业出版社，2010.

[3] 梁循. 数据挖掘算法与应用. 北京：北京大学出版社，2006.

[4] MargaretH.Dunham 著. 郭崇慧，田凤占，靳晓明等译. 数据挖掘教程. 北京：清华大学出版社，2005.

[5] 谢邦昌. 数据挖掘基础与应用. 北京：机械工业出版社，2011.

[6] 陈文卫，黄金才，赵新昱. 数据挖掘技术. 北京：北京工业大学出版社，2002.

[7] 陈京民等. 数据仓库与数据挖掘技术. 北京：电子工业出版社，2002.

[8] 焦李成，刘芳，缑水平等. 智能数据挖掘与知识发现. 西安：西安电子科技大学出版社，2006.

[9] 倪志伟，倪丽萍，刘慧婷，贾瑞玉. 动态数据挖掘. 北京：科学出版社，2010.

[10] 李雄飞，李军. 数据挖掘与知识发现. 北京：高等教育出版社，2003.

[11] 邵峰晶，于忠清，王金龙，孙仁诚等. 数据挖掘原理与算法. 北京：科学出版社，2009.

[12] 蒋盛益，李霞，郑琪等. 数据挖掘原理与实践. 北京：电子工业出版社，2011

[13] 蒋盛益主编. 商务数据挖掘与应用案例分析. 北京：电子工业出版社，2014.

[14] 吴昱. 大数据精确挖掘. 北京：化学工业出版社，2014.

[15] 陆旭. 文本挖掘中若干关键问题研究. 北京：中国科学技术大学出版社，2008.

[16] 洪文学，王金甲等. 可视化模式识别. 北京：国防工业出版社，2014.

[17] 赵刚著. 大数据技术与应用实践指南. 北京：电子工业出版社，2013.

[18] Danie T.Larose 著. 刘燕权，胡赛全等译. 数据挖掘方法与模型. 北京：高等教育出版社，2011.

[19] Ian H. Witten Eibe Frank Mark A.Hall 著. 李川，张永辉等译. 数据挖掘－实用机器学习工具与技术. 北京：机械工业出版社，2014.

[20]（加）洪松林、（中）庄映辉，李堃著. 数据挖掘技术与工程实践. 北京：机械工业出版社，2014.